

OPERATION MANUAL

Revision 4.05



TITAN-SVX SERVO MOTOR CONTROLLER-DRIVER



COPYRIGHT© 2017 ARCUS,
ALL RIGHTS RESERVED

First Edition, Jan 2017

ARCUS TECHNOLOGY copyrights this document. You may not reproduce or translate into any language in any form and means any part of this publication without the written permission from ARCUS.

ARCUS makes no representations or warranties regarding the content of this document. We reserve the right to revise this document any time without notice and obligation.

Table of Contents

1. INTRODUCTION	5
1.1. TECHNICAL FEATURES	6
2. GENERAL OPERATION OVERVIEW	7
2.1. PULSE MODE	7
2.2. CONTROL MODE	8
2.2.1. Position Value	8
2.2.2. Motion Profile	8
2.2.3. Homing	9
2.2.4. Limits	11
2.3. MOTOR POWER	11
2.4. JOG MOVE	11
2.5. STOPPING	12
2.6. POSITIONAL MOVES	12
2.7. MOTOR STATUS	12
2.8. FAULT STATUS	12
2.9. DIGITAL INPUTS / OUTPUTS	13
2.10. ANALOG INPUTS	14
2.11. JOYSTICK CONTROL	14
2.12. CLOSED LOOP CONTROL GAINS	15
2.12.1. P-Gain	15
2.12.2. V-Gain	15
2.12.3. I-Gain	15
2.12.4. C-Gain	15
2.13. DYNAMIC GAINS	15
2.14. FORCE CONTROL	16
2.15. STANDALONE PROGRAM SPECIFICATION	18
2.15.1. Standalone Program Specification	19
2.15.2. Multi-thread / multi-tasking	19
2.15.3. Standalone Subroutines	19
2.15.4. Standalone Variables	19
2.15.5. Math Operations	20
2.15.6. Standalone Run On Boot-Up	20
2.15.7. Storing Standalone Program to Flash	20
2.15.8. Standalone Command Set	21
2.15.9. Conditional Statements	24
2.15.10. Example Standalone Programs	25
3. SERIAL COMMUNICATING COMMANDS	29
3.1. STATUS COMMANDS	29
3.2. LED COMMANDS	29
3.3. MOTION COMMANDS	30
3.4. GAIN COMMANDS	30
3.5. STANDALONE PROGRAM COMMANDS	31
3.6. LIMIT COMMANDS	31
3.7. DIGITAL IO COMMANDS	32
3.8. VARIABLE COMMANDS	32
3.9. COMMUNICATION COMMANDS	32

3.10. FAULT MONITORING COMMANDS	33
3.11. MISCELLANEOUS COMMANDS	33
3.12. TREND MONITORING COMMANDS.....	34
3.13. FORCE CONTROL COMMANDS	34
4. SOFTWARE OVERVIEW	35
4.1. SOFTWARE INSTALLATION	35
4.1.1. <i>TITAN Directory</i>	37
4.2. COMMUNICATION AND SETUP REQUIREMENTS.....	38
4.3. STARTING THE TITAN-SVX UI SOFTWARE	39
4.4. COMMUNICATION.....	40
4.4.1. <i>USB/Serial Communication</i>	41
4.4.2. <i>Ethernet Communication</i>	44
4.5. MOTOR.....	45
4.5.1. <i>Motor Information</i>	46
4.5.2. <i>Position Sensor Parameters</i>	46
4.5.3. <i>Motor Electric Parameters</i>	47
4.5.4. <i>Motor Mechanical Parameters</i>	47
4.5.5. <i>Motor Database Wizard</i>	48
4.6. TUNING	57
4.6.1. <i>Mechanical Parameters</i>	59
4.6.2. <i>Mechanical Parameter Auto Detect Wizard</i>	60
4.7. CONFIGURATION	64
4.7.1. <i>Mechanical Parameters</i>	65
4.7.2. <i>Controller Mode Setup</i>	65
4.7.3. <i>Pulse Mode Setup</i>	66
4.7.4. <i>Special Configurations</i>	67
4.7.5. <i>Error Conditions / ESTOP Input</i>	67
4.7.6. <i>Analog Input Joystick Related Parameters</i>	68
4.7.7. <i>Position Related Parameters</i>	68
4.7.8. <i>Ethernet Socket Communication Settings</i>	69
4.7.9. <i>Power Up State</i>	69
4.8. PARAMETERS.....	70
4.9. TEST DRIVE.....	72
4.9.1. <i>Control Mode</i>	73
4.9.2. <i>Pulse Mode</i>	84

1. Introduction

The TITAN-SVX is an advanced single-axis closed loop servo driver-controller that supports various types of motors that are commonly used in the automation industry:

- 2 Phase Stepper Motor
- 3 Phase Brushless Rotary Servo Motor
- 3 Phase Brushless Linear Servo Motor
- DC Voice Coil Motor

In addition to the advanced servo motion control technology, the TITAN-SVX also has a number of advanced control technologies including force control, joystick control, dynamic gains, standalone programming, and many more. Voltage, current, temperature, and position monitoring allow for the TITAN-SVX to examine system trends and allow for preventative measures to reduce system down time.

TITAN-SVX is a true intelligent motion controller driver that enables and readies the future in the field of Smart Factory and Automation and Industrial Internet of Things.

1.1. Technical Features

- Communication using RS-485 multi-drop network:
 - 115200 bps, 8N1
 - Communication Protocol supported:
 - TITAN-ASCII
 - TITAN-ASCII with CRC
 - MODBUS-ASCII
 - MODBUS-RTU
- 100Mbps Ethernet communication using ASCII over TCP/IP¹
- USB communication using VCP
- Standalone programmable using Arcus A-SCRIPT language with support of 3 multi-thread programs
- Closed Loop Driver Specifications:
 - 24-48 VDC
 - 8.0 Amp max peak current setting
 - 1 MHz max pulse support (in Pulse Mode)
- Multiple types of motor support:
 - 2 Phase Bipolar Stepper Motors
 - 3 Phase Brushless Rotary Servo Motors
 - 3 Phase Brushless Linear Servo Motors
 - DC Voice Coil Motors
- Configurable in following modes:
 - Pulse Mode - digital pulse control using pulse/dir or CW/CCW
 - Control Mode – internal motion profile generation with motion sequence control from internal standalone programming.
- Opto-isolated Digital IO:
 - 8 bits of digital inputs
 - 3 bits of digital outputs
- 1 x 12-bit analog input
 - Joystick control
- A/B/Z differential encoder inputs with A/B/Z single ended encoder signal outputs
- UVW Hall sensor digital inputs
- Control Mode Features:
 - Homing routines using combination of Home/Limit/Z Index
 - Soft and Hard Limit Protection.
 - Over-current/Over-voltage/ Under voltage/Temperature/Position Error fault detection
 - Force/Torque Control
 - Dynamic gains

¹ Only applicable to TITAN-SVX-ETH model

2. General Operation Overview

All commands described in this section refer to ASCII commands sent over RS485, USB, or Ethernet. Corresponding standalone commands are details in the Standalone Command Set section. A full list of valid ASCII commands can be found in section 3.0.

The TITAN-SVX can be used in Pulse Mode or Control Mode.

Pulse Mode is used when the control position signals are sent from an external motion controller that generates pulse and direction signals (or CW/CCW signals). In Pulse Mode, standalone programming is supported but motion related commands are not supported.

Control Mode is used when the motion control and motion profiling is done internally by the TITAN-SVX controller. Control Mode supports standalone programming with all motion commands.

2.1. Pulse Mode

In order to use the controller as pulse mode, TITAN-SVX must be configured as Pulse Mode. Please refer to the Configuration section of the TITAN-SVX Windows UI program to set the controller to the Pulse Mode.

In Pulse Mode, the following digital inputs and outputs are used:

INPUTS:

- Pulse (CW) – sets the target position of the motion.
- Dir (CCW) – sets the direction of the motion
- Enable – enables the servo power to the motor
- Clear – clears any fault
- Reset – resets the internal position counter to zero

OUTPUTS:

- In Position – when the difference in target and actual position value is within the in-position amount, this output turns on.
- Alarm – when the TITAN-SVX goes into error (such as position range error, stall current error, over temperature error, etc.) this output turns on. To clear the alarm state, toggle the Clear input or the Enable input.

Polarity of the inputs and outputs can be configured as active low or active high. Please refer to the Configuration section of the manual for various settings.

2.2. Control Mode

In order to use the controller in Control Mode, TITAN-SVX must be configured as Control Mode. Please refer to the Configuration section of the TITAN-SVX- Windows UI program to set the controller to the Control Mode.

In Control Mode, following digital inputs are used:

INPUTS:

- +LIM (DI6) – Plus Limit Input
- HOME (DI7) – Home input
- -LIM (DI8) – Minus limit input

All other remaining digital inputs and outputs (5 digital inputs and 3 digital outputs) are available as general purpose use.

In Control Mode, the limit inputs can be disabled and used as general purpose inputs.

2.2.1. Position Value

By default, the position values of TITAN-SVX are in encoder counts. The **EX** command can be used to query the current encoder position. The user must convert the encoder or pulse counts to actual position unit such as mm, inch, or degrees.

For example, consider TITAN-SVX driving a stepper motor with 16,000 encoder counts per rev that is connected to a ball screw linear drive with pitch of 10mm. In this linear system, 1mm move will be 1,600 encoder count of the motor.

During operation, the current target position of the TITAN-SVX can be read using the **POSD** command. Additionally the current position error can be read using the **PERR** command. The response of both commands will be in units of encoder counts.

2.2.2. Motion Profile

The TITAN-SVX uses a trapezoidal velocity profile as shown in Figure 2.0.

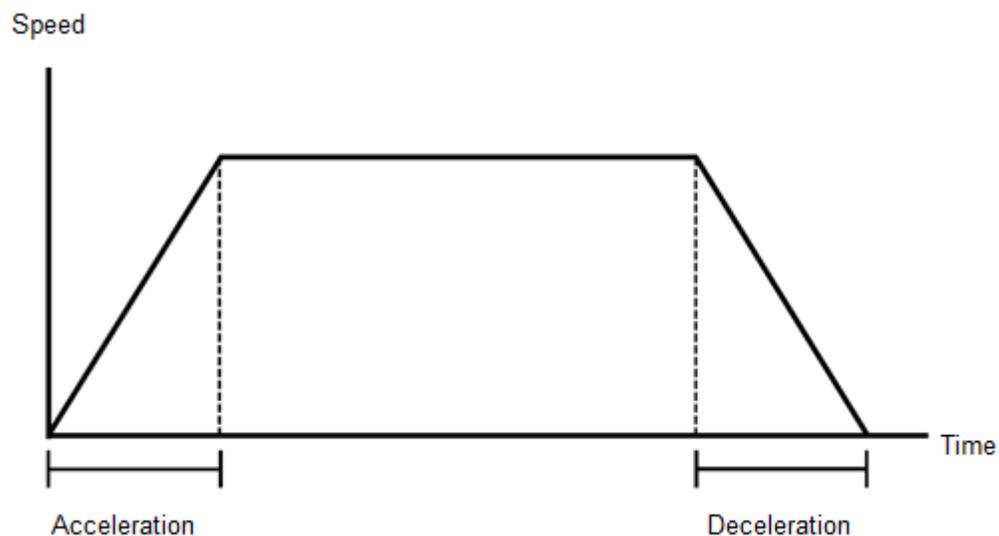


Figure 2.0

Target speed can be set using the **HSPD** command. The speed value is in either in **RPM** or **mm/sec** depending on whether the type of motor is configured as rotary or linear. Note that all rotary motors are configured as **RPM** and all linear motors are configured as **mm/sec** speed value.

Speed parameters can be changed to **encoder counts/sec** in order to achieve slower motion. See the configuration section of the TITAN-SVX software in section 4 to make this setting. The lowest allowable speed setting will be 1. Depending on the mode, this will correspond to **1 RPM (mm/sec)** or **1 encoder count/sec**.

Target acceleration can be set using the **ACC** command. Acceleration values are in either **RPM/sec** or **mm/sec²** depending on the setting as rotary or linear. Acceleration represents the rate of increase of the speed, or the slope of the speed/time chart as shown in figure 6.0. Note that the higher the acceleration value, the steeper the speed/time graph (slope) will be and quicker to reach the target speed.

For short duration profiles where the speed cannot be reached, the controller automatically uses a triangle profile, and high speed will not be reached.

Acceleration value is used for both acceleration and deceleration of the motion profile.

2.2.3. Homing

The TITAN-SVX has various homing routines using the home, limit, and encoder index inputs. To perform a homing routine, the homing mode must be selected using the **HMODE** command. See table 2.0 for details.

Value	Description
0	Positive direction using home sensor
1	Negative direction using home sensor
2	Positive direction using positive limit sensor
3	Negative direction using negative limit sensor
4	Positive direction using home sensor and following encoder index
5	Negative direction using home sensor and following encoder index
6	Positive direction using positive limit sensor and following encoder index
7	Negative direction using negative limit sensor and following encoder index
8	Positive direction using encoder index
9	Negative direction using encoder index

Table 2.0

Once a homing mode is selected, the **HOMEX** command will perform the homing routine.

Hard stop homing is also available by using a standalone program that senses the motor current or position error. A sample standalone program is included in the TITAN-SVX software installation.

The following example shows the example of home search in positive direction using the home input plus encoder index input.

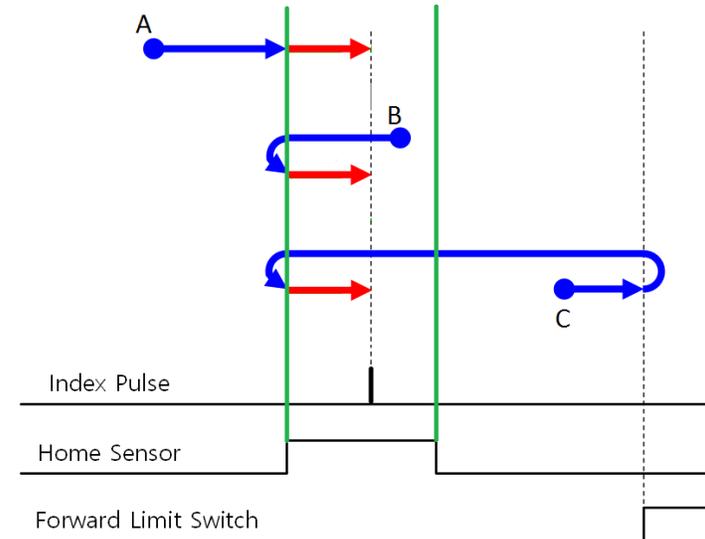


Figure 2.1

In the homing example in figure 2.1, the following are the points where the homing can start:

- A. Homing is initiated in positive direction and rising edge of the home signal is detected. Index is searched as soon as the rising edge of the home signal is detected.
- B. Homing starts in home input on state. The homing routine moves the motor out of the home sensor and search back to find the rising edge of the home sensor. Encoder index sensor is searched as soon as the rising edge of the home sensor is detected
- C. Homing starts in positive direction and positive limit switch is detected. Motion moves out of the limit switch and searches for the rising edge and then falling edge of the home sensor and then moves back to find the rising edge of the home sensor. Encoder index searched as soon as the rising edge of the home sensor is detected.

The above example shows the homing routine that will always find a consistent home position.

2.2.4. Limits

TITAN-SVX has both software and hardware limit monitoring.

Software limit function can be enabled and disabled using the **SLIMON** command. The positive and negative software limit position values can be set in the Configuration Section or by using the **SLIMPOS** and **SLIMNEG** commands.

The hardware limit can be enabled and disabled in the Configuration section.

The following actions are available when either soft limit or hard limit is triggered. Action is determined by the value of the **LIMPRO** command.

Value	Description
0	Disable Motor
1	Immediate stop with servo on

Table 2.1

When limit function is enabled and the limit input is triggered during motion, the motion state goes into an error state. If the standalone program is running and error occurs, the program state goes into error state. Error state must be cleared before any further motion can be performed or program can be run.

2.3. Motor Power

The motor can be enabled or disabled during operation. The **SVON** command will enable the motor and the **SVOFF** command will disable the motor. By default, the TITAN-SVX will power on with the motor disabled. Settings in the Configuration section of the TITAN-SVX GUI allow for the motor to be enabled on power up.

If a stepper motor is being used, an open loop hold option is available to keep the motor steady while idle. Use the **OLPHOLD** command to activate this feature. When enabled, the motor will not servo while idle. This option is not available for BLDC or DC motors.

Value	Description
0	Deactivated. Full time closed loop servo
1-100	Percentage current hold when enabled and idle

Table 2.2

2.4. Jog Move

A jog move is used to continuously move the motor without stopping. The **JOGXP** command will jog the motor in the positive direction and the **JOGXN** command will jog the motor in the negative direction. Once this move is started, the motor will only stop if a limit condition is activated during motion or a stop command is issued.

If a motion command is sent while the controller is already moving, the command is not processed. Instead, an error response is returned.

2.5. Stopping

When the motor is performing any type of move, motion can be stopped using the **STOPX** command. The stop command will use the value in the **ACC** command as the deceleration for the stop.

2.6. Positional Moves

The TITAN-SVX can perform moves to a defined position. The target position will be in terms of encoder counts. Use the command **X=[target]** to perform a positional move. For example, the **X=1000** command will move the motor to encoder position 1000.

If a motion command is sent while the controller is already moving, the command is not processed. Instead, an error response is returned.

2.7. Motor Status

Motor status can be read anytime by reading the response to the **MST** command. The following is the bit representation of motor status.

Bit	Description
0	Enabled
1	In Position
2	Moving
3	In Fault

Table 2.3

2.8. Fault Status

If the motor is in a fault condition, as indicated by the **MST** command, the fault status can be read from the response to the **FLT** command. The following is the bit representation of fault status.

Bit	Description
0	Negative Limit Error
1	Positive Limit Error
2	Position Error
3	Current Error
4	Hall Sensor Error
5	Over Current Error
6	Over Voltage Error
7	Under Voltage Error
8	Over Temperature Error
9	Motor Connection Error
10	Emergency Switch On
11	Encoder Error

Table 2.4

The **ECLEARX** command can be used to clear a fault condition.

2.9. Digital Inputs / Outputs

TITAN-SVX module comes with 8 digital inputs and 3 digital outputs. Depending on the mode of the TITAN-SVX, Pulse Mode or Control Mode, digital inputs and outputs are used in various ways.

Pulse Mode	Control Mode	Pulse Mode Description	Control Mode Description
PUL(CW)	DI1	Pulse Input	Digital Input 1
DIR(CCW)	DI2	Dir Input	Digital Input 2
ENA	DI3	Enable Input	Digital Input 3
CLR	DI4	Clear Fault Input	Digital Input 4
RST	DI5	Reset Input (EMG)	Digital Input 5 (EMG)
DI6	+LIM	Digital Input 6 (Not Used)	+ Limit Input
DI7	HOME	Digital Input 7 (Not Used)	Home Input
DI8	-LIM	Digital Input 8 (Not Used)	- Limit Input
ALM	DO1	Alarm Output	Digital Output 1
DO2	DO2	Digital Output 2 (Not Used)	Digital Output 2
INP	DO3	In Position Output	Digital Output 3

Table 2.5

The **DIN** command will return the status of all 8 available digital inputs in a bitwise assignment.

The **DOUT** command can be used to set or read the value of all 3 digital outputs in a bitwise assignment.

+L and -L can be disabled as limit switch input and can be used as general purpose inputs. Check the Configuration section of the Windows program.

In Pulse Mode, PUL and DIR (CW/CCW) are used for position input and ENA is used for enabling and disabling the power to the motor.

CLR (or DI4) is used for clearing any faults. RST (or DI5) is used to reset the position and error. DI5 can be configured as Emergency input. When EMG is enabled, the triggering of DI5 will disable the motor immediately.

DI6, DI7, and DI8 are not used in Pulse mode.

In Pulse Mode, there are 3 digital outputs. ALM is turned on when there is any fault in the driver. INP is turned on if the target position is within the in-position tolerance value. Digital output 2 is a general purpose digital input.

Polarity of the digital outputs and inputs can be configured to be active high or active low. See Configuration section of the software, detailed in section 4.

2.10. Analog Inputs

The **AIN** command will retrieve the current 5V 12-bit analog input status of the TITAN-SVX. The return value can range from 0 to 4095. A value of 0 will correspond to 0V and 4095 will correspond to 5V.

2.11. Joystick Control

A joystick control is available using the 12-bit analog input. The analog input value can range from 0 to 4095.

The joystick operation will move the motor in the positive or negative direction based on the value of the analog input. The maximum allowable speed of the joystick operation is defined by the **AJMS** command.

Parameters for the low deadband value and the high deadband value will be defined by the **AJDBL** and **AJDBH** commands respectively. While the analog input is between these two values, joystick operation will not move the motor. The bandwidth parameter, defined by the **AJBW** command, will determine the slew rate of the motor as the analog input increases or decreases beyond the deadband zone.

The relationship between the deadband zone and bandwidth can be seen in the figure below.

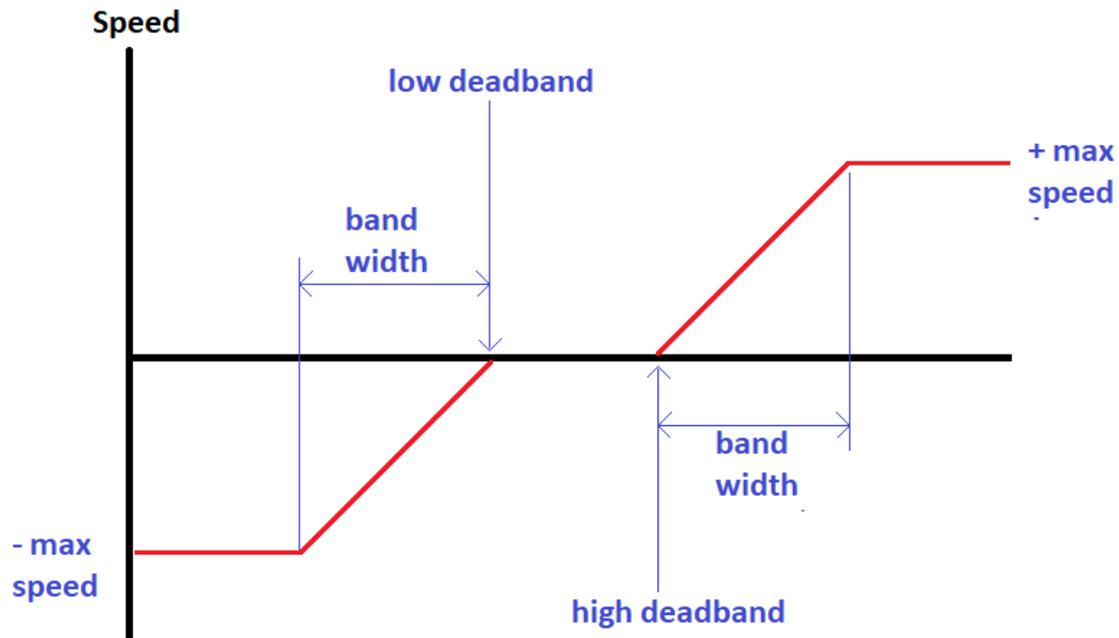


Figure 2.2

The soft limits for joystick control can be implemented using the standard soft limit feature.

2.12. Closed Loop Control Gains

TITAN-SVX has the following simplified gain settings available.

- P Gain – Position gain.
- V Gain – Velocity gain.
- I Gain – integral gain
- C Gain – current gain

All the gains have range from 0 to 100. The simplified gain values are derived using special formulas that are converted to normalized values ranging from 0 to 100. Value of 0 represents weak gain value and 100 represents strong gain values.

2.12.1. P-Gain

P-Gain is also known as position gain. Position gain determines the firmness of the closed loop control. P-Gain is similar to Proportional gain in the PID Loop. Use the **PGAINF** command to access the P-Gain.

2.12.2. V-Gain

V-Gain is also known as velocity gain. Velocity gain is mainly used to reduce the velocity error and mainly used to reduce overshoots in the vibration. V-Gain is similar to Derivative gain in the PID Loop. Use the **VGAINF** command to access the V-Gain.

2.12.3. I-Gain

I-Gain is also known as integral gain. Integral gain is mainly used to reduce steady state position error. I-Gain is similar to Integral gain in the PID Loop. Use the **IGAINF** command to access the I-Gain.

2.12.4. C-Gain

C-Gain is also known as current gain. Current gain is mainly used for the current control and consists mainly of proportional and integral gain of the current control. Use the **CGAINF** command to access the C-Gain.

2.13. Dynamic Gains

Dynamic gain can be used to change the behavior of the closed loop servo depending on motor speed. Applications may require a higher responsiveness from the servo as the speed of the motor increases.

A low speed gain and a high speed gain can be defined using the **DGLGAIN** and **DGUGAIN** respectively. The low speed used for the low speed gain setting can be set using the **DGLSPD** command. The high speed used for the high speed gain setting can be set using the **DGUSPD** command.

There is a linear relationship between the low and high speed gains as the speed transitions from low to high speed. See the graph in figure 2.3 for details.

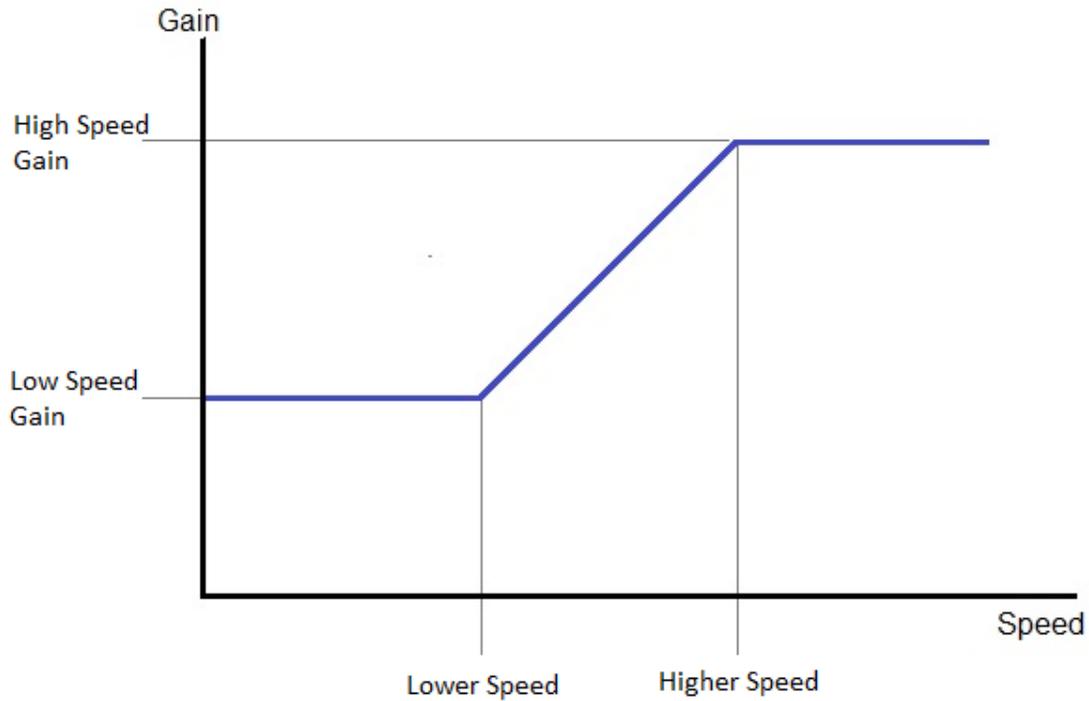


Figure 2.3

Dynamic gain can also be based on the actual or target velocity. The **DGTYPE=0** command will use the profile velocity and the **DGTYPE=1** command will use the actual velocity.

The **DGENA** command can be used to enable (1) or disable (0) the dynamic gain feature.

2.14. Force Control

Force, or torque, generated by an electric motor is equal to the motor constant K_t multiplied by the motor current as represented by the following equation.

$$\text{Force (or Torque)} = K_t * \text{Current}$$

By monitoring and controlling the motor current, the TITAN-SVX is able to perform various types of force control operations. Titan-SVX has a built-in force control routine with the following steps.

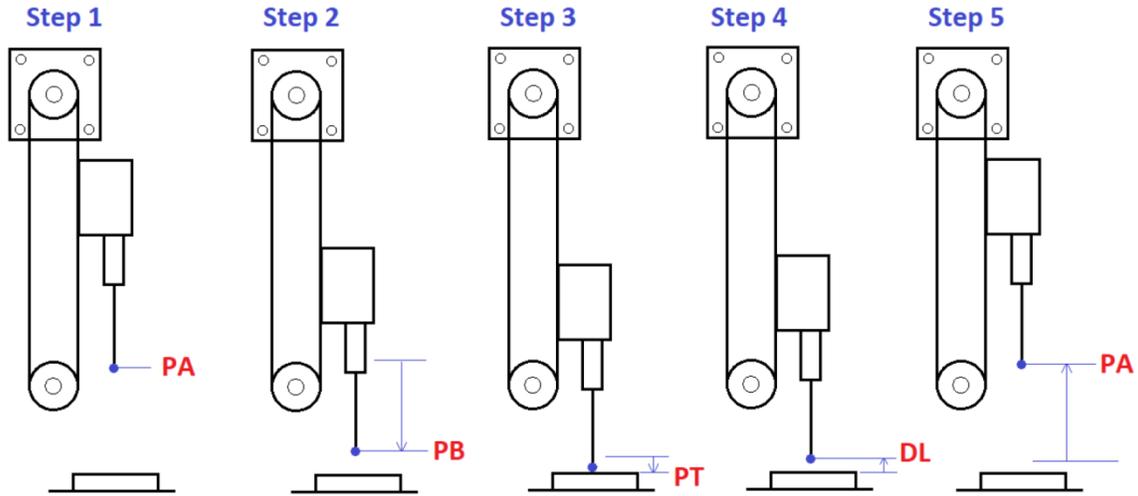


Figure 2.4

Step	Description	Relevant Variables
1	Force control starts from initial position (FCPA). If not in this position at the start of the routine, the motor is moved to this location at high speed (FCVC) and acceleration (FCAC).	FCPA – Initial Position FCVC – Velocity to PA FCAC – Acceleration to PA
2	Motor is moved to the start of search position (FCPB) at high speed (FCVA) and acceleration (FCAA). FCCA defines the max current used for the move to FCPB .	FCPB – Start Search Position FCVA – Velocity to PB FCAA – Acceleration to PB FCCA – Maximum current used moving to PB
3	From FCPB position, the motor is decelerated to velocity FCVB and moved until the current trigger is detected, at position FCPT . Once the trigger is detected, the following two force operations are performed in parallel: <ul style="list-style-type: none"> Decelerate to a stop from the detection point or move to position FCDP further down maintaining the force. Move further down is performed in case the detected point is pushed down and the force needs to be maintained. Countdown timer is started in order to maintain force for the set duration amount. 	FCPT – Current trigger position value FCVB – Velocity moving to FCPT FCAB – Deceleration used after finding FCPB FCCB – Maximum current used during the search FCCC – Trigger current to detect FCPT FCCF – Current used to apply continuous force after detecting FCPT FCDP – Additional delta position move from the trigger position FCPT for applying continuous force. Motor may not move during FCDP move. FCPM – Maximum position to move before determining trigger not found.
4	Once the continuous force operation and countdown timer operation are satisfied, the motor is moved away from the current position by the position amount FCDL . High speed FCVB and acceleration FCAB are used.	FCVB – Velocity used for lift move FCAB – Acceleration used for lift move FCDL – Lift movement amount from current location FCCF – Current used during lift move
5	Motor is moved back to the initial position FCPA at high speed (FCVC) and acceleration (FCAC).	FCPA – Initial Position FCVC – Velocity to PA FCAC – Acceleration to PA

Table 2.6

The Force Control Mode command **FCMODE** allows for the following force control options.

Bit	Description
0	Filter Current – while the motor is moving in search of the current trigger position, the moving average filter can be enabled to reduce noise in the motor current reading. Note that enabling filter current will reduce the sensitivity.
1	Wait Acknowledge – when the motor current trigger point is detected, the acknowledge command is expected before performing the lift move and moving back to the initial position. The acknowledge command can come from either the host controller or by digital input.
2	DIO Control – Digital inputs and output are used for the Force Control.

Table 2.7

Digital input and outputs can be used for the Force Control routine. To use digital IO for the Force Control, enable bit 2 of the force control mode parameter **FCMODE**.

See the table below for the digital IO assignment in regards to the Force Control feature.

I/O	Description
DI1	Start the force control routine (active high). Also used to issue acknowledge command (falling edge) if enabled by the FCMODE parameter.
DI2	Abort the force control routine (active high). When triggered during the force control routine, the motor is immediately moved back to the initial position FCPA using FCVC and FCAC .
DO1	Force control routine status. High signal indicates the force control routine is in progress. Low indicates the force control routine is completed.
DO2	Waiting for acknowledge. High signals means that force trigger is detected and waiting for the acknowledge signal.
DO3	Found trigger. High signal means the force trigger is found.

Table 2.8

2.15. Standalone Program Specification

Standalone programming allows the controller to execute a user defined program that is stored in the internal memory of the TITAN-SVX. The standalone program can be run independent of host communication.

Standalone programming can be used in both Pulse Mode and Control Mode. Note that in Pulse Mode, motion commands are not supported since the motion is controlled using the digital inputs.

Standalone programs are written, compiled, and downloaded to the TITAN-SVX using the GUI program. Standalone program also can be uploaded from TITAN-SVX.

The standalone program can be stored in the flash and auto start can be enabled so that the standalone program run on controller power up.

The TITAN-SVX supports up to three separate standalone programs simultaneously which is also known as multi-threading or multi-tasking. For example, program 0 can be used to perform motion related functions. Program 1 can be used for digital IO monitoring and triggering. Program 2 can be used for dynamic gain control.

2.15.1. Standalone Program Specification

Memory size: max total 1000 standalone lines (all 3 programs combined).

2.15.2. Multi-thread / multi-tasking

The TITAN-SVX supports the simultaneous execution of all three standalone programs.

2.15.3. Standalone Subroutines

The TITAN-SVX has the capability of using up to 32 separate subroutines. Subroutines are typically used to perform functions that are repeated throughout the standalone program. Note that subroutines can be shared by multi-threads of standalone programs.

Standalone programs can jump to subroutine in the standalone program. The subroutines are referenced by their subroutine number [SUB 1 - SUB 32]. If a subroutine number is not defined, the controller will return with an error.

2.15.4. Standalone Variables

The TITAN-SVX has 100 32-bit signed integer standalone variables available for general purpose use. They can be used to store values, perform basic calculations, and support integer operations. Floating point values are not supported. Note that in the math operations, floating point calculation is done but the values are saved as integer whole number values.

Any read commands, variables, or constant values can be assign to a variable. Variables are initialized to zero on power up. Table 2.9 are values that can be assigned to a variable

Argument	Description	Example
[CONST]	Constant Value	V1=200
V[var num]	Variable	V1=V2
MSTX	Motor Status	V1=MSTX
PROBEST	Probe Status	V1=PROBEST
CUR	Motor Current	V1=CUR
ERX	Position Error	V1=ERX
PX	Profile Position	V1=PX
EX	Actual Encoder Position	V1=EX
SPDX	Actual Speed	V1=SPDX
DI	Digital Inputs (bit 0-7)	V1=DI
DI[val]	Digital Input Bit	V1=DI2
DO	Digital Outputs (bit 0-2)	V1=DO
DO[val]	Digital Output Bit	V1=DO1

Table 2.9

2.15.5. Math Operations

The following math operations are available for use in the standalone programming:

Operator	Description	Example
+	Integer Addition	$V1=V1+2$
-	Integer Subtraction	$V1=V2-V3$
*	Integer Multiplication	$V1=V2*200$
/	Integer Division (round down)	$V1=V3/2$
%	Modulus	$V1=V2\%5$
>>	Bit Shift Right	$V1=V2>>2$
<<	Bit Shift Left	$V1=V2<<2$
&	Bitwise AND	$V1=V2\&7$
	Bitwise OR	$V1=V2 8$
~	Bitwise NOT	$V1=\sim V2$

Table 2.10

2.15.6. Standalone Run On Boot-Up

Standalone programs can be configured to run on controller power up. Refer to the Windows program on Test Drive window.

Important Note: When running on power up, be careful that the program does not move the motor in uncontrolled manner and damage the system.

2.15.7. Storing Standalone Program to Flash

Standalone programs need to be stored on the flash of the TITAN-SVX, which allows the program to be preserved after powering down the controller. This is also required if the program is to be started from power up.

Important Note: When standalone programs (or parameters) are stored to flash memory, the motor servo will be turned off.

Program with comments can also be stored to the flash. Please refer to the Test Drive section of the software, detailed in section 4.

2.15.8. Standalone Command Set

The command set for standalone programming is detailed in table 2.11. These commands are only valid when using the standalone programming feature. The ASCII commands sent over RS-485, USB, or Ethernet are described section 3.

Command	Description		Example
;	Comment notation. Comments out any text following ";" in the same line.		;***This is a comment
ACC	Set the acceleration setting. Unit is in RPM/sec or mm/sec/sec depending on the type of motor is configured.	W	ACC=5000 ACC=V1
CGAIN	Sets the Current Gain. Range is 0 to 100	W	CGAIN=80
CUR	Gets the motor current	R	V1=CUR IF CUR>500
DELAY	Set a delay in milliseconds. Assigned value is a 32-bit unsigned integer or a variable.	W	DELAY=1000 ;1 second DELAY=V1 ;assign to variable
DI	Return status of digital inputs. Bit 0 – DI1 (Pulse – Pulse Mode) Bit 1 – DI2 (Dir – Pulse Mode) Bit 2 – DI3 (Enable – Pulse Mode) Bit 3 – DI4 (Clear – Pulse Mode) Bit 4 – DI5 (Reset – Pulse Mode) Bit 5 – DI6 (+Lim – Control Mode) Bit 6 – DI7 (Home – Control Mode) Bit 7 – DI8 (-Lim – Control Mode)	R	V2=DI IF DI=0 DO=1 ;Turn on DO1 ENDIF
DI[1-8]	Get individual bit status of digital inputs. Will return [0,1].	R	IF DI1=0 DO=1 ;Turn on DO1 ENDIF V3=DI1
DO	Set/get digital output status. Bit 0 – DO1 (Alarm – Pulse Mode) Bit 1 – DO2 Bit 2 – DO3 (InPos – Pulse Mode)	R/W	DO=2 ;Turn on DO2 V1=DO
DO[1-3]	Set/get individual bit status of digital outputs. Value range is [0,1].	R/W	DO2=1 ;Turn on DO2 V1=DO3
ECLEARX	Clears motion error state	W	
END	Indicate end of program. Program status changes to idle when END is reached. NOTE: Subroutine definitions should be written AFTER the END statement.		
ENDSUB	Indicated end of subroutine. When ENDSUB is reached, the program returns to the previously called subroutine.		
EX	Set/get the current encoder position.	R/W	EX=1000 EX=V1 V1=EX
ERRORCHECK	Sets the error checking Bit 0 – Position Error Check Bit 1 – Stall Current Error Check	W	ERRORCHECK=1 ERRORCHECK=V2
ERX	Reads position error	R	V1=ERX
GOSUB [1-16]	Call a subroutine that has been previously stored to flash memory.		GOSUB 1
HMODE	Sets the homing mode: 0 – Plus Home 1 – Minus Home 2 – Plus Limit 3 – Minus Limit 4 – Plus Home Plus Encoder Index	W	HMODE=1 HMODE=V1 HOMEX WAITX

	5 – Minus Home Plus Encoder Index 6 – Plus Limit Plus Encoder Index 7 – Minus Limit Plus Encoder Index 8 - Plus Index Home 9 - Minus Index Home		
HOMEX	Home search the motor using the HMODE setting	W	HMODE=0 HOMEX
HSPD	Set/get the global high speed setting. Unit is in pulses/second.	W	HSPD=1000 HSPD=V1
IF ELSEIF ELSE ENDIF	Perform a standard IF/ELSEIF/ELSE conditional. Any command with read ability can be used in a conditional. ENDIF should be used to close off an IF statement. Conditions [=, >, <, >=, <=, !=] are available		IF DI1=0 DO=1 ELSEIF DI2=0 DO=2 ELSE DO=0 ENDIF
IGAIN	Sets the Integral Gain. Range is 0 to 100	W	IGAIN=30 IGAIN=V30
INPOS	Set/Get In-position range value	W	INPOS=10 INPOS=V1
JOGX+	Jogs the motor in positive direction	W	JOGX+
JOGX-	Jogs the motor in negative direction	W	JOGX-
LED	Sets the LED light status. 0 – turn off LED 1 – Turn on LED	W	LED=1 LED=0
LIMITCHECK	Set/Get Limit Checking Bit 0 – Enable Hard Limit Checking Bit 1 – Enable Soft Limit Checking	W	LIMITCHECK=0 LIMITCHECK=V1
MAXCUR	Sets maximum current limit	W	MAXCUR=2500 ;Sets 2.5A MAXCUR=V24
MSTX	Gets motor status Bit 0 – Enabled Bit 1 – Motor in Fault State Bit 2 – Motor moving Bit 3 – Motor Homing Bit 4 – Reserved Bit 5 – Reserved Bit 6 – Reserved Bit 7 – Soft Limit Triggered Bit 8 – In Position Bit 9 – Homing Complete	R	V1=MSTX
PGAIN	Sets the Position Gain. Range is 0 to 100	W	PGAIN=30 PGAIN=V30
PRESETGAIN	Sets all the Gains (PGAIN, VGAIN, IGAIN) to same value. Range is 0 to 100	W	PRESETGAIN=30 PRESETGAIN=V30
PROBEAM PROBEBM PROBECM	Selects data type for Probe A/B/C Mode 0 – none 1 – Actual Position 2 – Actual Velocity 3 – Target Position 4 – Target Velocity 5 – Position Error 6 – Velocity Error 7 – Actual Current 8 – Target Current	W	PROBEAM=1 PROBEBM=2 PROBECM=5

PROBECMD	Sets the probe command 1 – start collecting probe data 2 – stop collecting probe data	W	PROBECMD = 1 ;***Start probe PROBECMD = 2 ;***Stop probe
PROBEDT	Probe Delta Time – Sets the time between the probe. Value is in msec.	W	PROBEDT=5 ;***5 msec probe
PGAIN	Sets the Position Gain. Range is 0 to 100	W	PGAIN=65 PGAIN=V2
PRG[0-1]	Used to define the beginning and end of a main program. Two standalone programs are available.		PRG 0
PX	Get the current motor target demand position.	R	V1=PX IF PX>100
RGB	Sets internal Red/Green/Blue LED	W	RGB=7
SPDX	Gets current speed	R	V1=SPDX
SR[0-2]	Run or stop the standalone program. 0 – stop 1 – run	W	SR0=0 ;Turn off program 0
STOPX	Stop motion using a decelerated stop.	W	STOPX
SVOFF	Disables the servo	W	SVOFF
SVON	Enables the servo	W	SVON
SUB [1-16] ENDSUB	Defines the beginning of a subroutine. ENDSUB should be used to define the end of the subroutine.		SUB 1 DO=4 ENDSUB
V[1-100]++	Increments Variable Value	W	V1++
V[1-100]--	Decrements Variable Value	W	V1--
V[1-100]	Set/get standalone variables. The following operations are available: [+] Addition [-] Subtraction [*] Multiplication [/] Division [%] Modulus [>>] Bit shift right [<<] Bit shift left [&] Bitwise AND [] Bitwise OR [~] Bitwise NOT	R/W	V1=12345 ;Set V1 to 12345 V2=V1+1 ;Set V2 to V1 + 1 V3=DI ;Set V3 to DI V4=DO ;SET V4 TO DO V5=~V1 ;SET V5 TO NOT V1
VGAIN	Sets the Velocity Gain. Range is 0 to 100	W	VGAIN=25 VGAIN=V24
WAITX	Wait for current motion to complete before processing the next line.	W	X1000 ;MOVE TO POSITION 1000 WAITX ;wait for move to finish
WHILE ENDWHILE	Perform a standard WHILE loop within the standalone program. ENDWHILE should be used to close off a WHILE loop. Conditions [=, >, <, >=, <=, !=] are available.		WHILE 1=1 ;FOREVER LOOP DO=1 ;Turn on DO1 DO=0 ;Turn off DO1 ENDWHILE
X[position]	If in absolute mode, move the X motor to [position]. If in incremental mode, move the motor to [current position] + [position].	W	X1000

Table 2.11

2.15.9. Conditional Statements

Standalone programming supports the following conditional statements:

```
IF / ENDIF
IF / ELSE / ENDIF
IF / ELSEIF / ENDIF
WHILE / ENDWHILE
```

The following conditions can be used in the conditional statement:

Condition	Description
>	Greater Than
<	Less Than
=	Equal
!=	Not Equal

Table 2.12

Following are values that can be used in the conditions

Argument	Description	Example
[CONST]	Constant Value	V1=200
V[var num]	Variable	V1<V2
MSTX	Motor Status	MSTX!=1
CUR	Motor Current	CUR>2000
ERX	Position Error	ERX>V1
EX	Actual Encoder Position	EX<10000
SPDX	Actual Speed	SPDX<1000
DI	Digital Inputs (bit 0-7)	DI=4
DI[val]	Digital Input Bit	DI2=1
DO	Digital Outputs (bit 0-2)	DO=3
DO[val]	Digital Output Bit	DO1=0

Table 2.13

Example:

```
IF V1 = 1
ENDIF

IF EX > 1000
ELSEIF SPDX<100
ENDIF

WHILE DI > V2
    V1=V1-1
ENDWHILE
```

2.15.10. Example Standalone Programs

Standalone Example Program 1 – Single Thread

Task: Set the speed and move the motor to 1000 and back to 0.

```
HSPD=200      ;* Set the high speed to 200RPM
ACC=3000     ;* Set the acceleration to 3000 RPM/sec
EO=1         ;* Enable the motor power
X1000        ;* Move to 1000
X0           ;* Move to 1000
END          ;* End of the program
```

Standalone Example Program 2 – Single Thread

Task: Move the motor back and forth indefinitely between position 1000 and 0.

```
HSPD=200      ;* Set the high speed to 200RPM
ACC=3000     ;* Set the acceleration to 3000 RPM/sec
EO=1         ;* Enable the motor power
WHILE 1=1    ;* Forever loop
    X1000    ;* Move to zero
    X0       ;* Move to 1000
ENDWHILE     ;* Go back to WHILE statement
END
```

Standalone Example Program 3 – Single Thread

Task: Move the motor back and forth 10 times between position 1000 and 0.

```
HSPD=200      ;* Set the high speed to 200RPM
ACC=3000     ;* Set the acceleration to 3000RPM/sec
EO=1         ;* Enable the motor power
V1=0         ;* Set variable 1 to value 0
WHILE V1<10 ;* Loop while variable 1 is less than 10
    X1000    ;* Move to zero
    X0       ;* Move to 1000
    V1=V1+1 ;* Increment variable 1
ENDWHILE     ;* Go back to WHILE statement
END
```

Standalone Example Program 4 – Single Thread

Task: Move the motor back and forth between position 1000 and 0 only if the digital input 1 is turned on.

```

HSPD=200           ;* Set the high speed to 200 pulses/sec
ACC=300            ;* Set the acceleration to 300 RPM/sec
SVON               ;* Enable the motor power
WHILE 1=1          ;* Forever loop
    IF DI1=1       ;* If digital input 1 is on, execute the statements
        X1000     ;* Move to zero
        X0        ;* Move to 1000
    ENDIF
ENDWHILE           ;* Go back to WHILE statement
END

```

Standalone Example Program 5 – Single Thread

Task: Using a subroutine, increment the motor by 1000 whenever the DI1 rising edge is detected.

```

HSPD=200           ;* Set the high speed to 200RPM
ACC=300            ;* Set the acceleration to 300 RPM/sec
SVON               ;* Enable the motor power
V1=0               ;* Set variable 1 to zero
WHILE 1=1          ;* Forever loop
    IF DI1=1       ;* If digital input 1 is on, execute the statements
        GOSUB 1    ;* Move to zero
    ENDIF
ENDWHILE           ;* Go back to WHILE statement
END

SUB 1
    XV1            ;* Move to V1 target position
    V1=V1+1000    ;* Increment V1 by 1000
    WHILE DI1=1   ;* Wait until the DI1 is turned off so that
    ENDWHILE      ;* 1000 increment is not continuously done
ENDSUB

```

Standalone Example Program 6 – Single Thread

Task: If digital input 1 is on, move to position 1000. If digital input 2 is on, move to position 2000. If digital input 3 is on, move to 3000. If digital input 5 is on, home the motor in negative direction. Use digital output 1 to indicate that the motor is moving or not moving.

```
HSPD=200          ;* Set the high speed to 200 pulses/sec
ACC=300           ;* Set the acceleration to 300 RPM/sec
SVON              ;* Enable the motor power
WHILE 1=1         ;* Forever loop
    IF DI1=1      ;* If digital input 1 is on
        X1000    ;* Move to 1000
    ELSEIF DI2=1 ;* If digital input 2 is on
        X2000    ;* Move to 2000
    ELSEIF DI3=1 ;* If digital input 3 is on
        X3000    ;* Move to 3000
    ELSEIF DI5=1 ;* If digital input 5 is on
        HOMEX-   ;* Home the motor in negative direction
    ENDIF
    V1=MSTX      ;* Store the motor status to variable 1
    V2=V1&7     ;* Get first 3 bits
    IF V2!=0
        DO1=1
    ELSE
        DO1=0
    ENDIF
ENDWHILE         ;* Go back to WHILE statement
END
```

Standalone Example Program 7 – Multi Thread

Task: Program 0 will continuously move the motor between positions 0 and 1000. Simultaneously, program 1 will monitors the status of digital input 1 and performs action depending on the status of digital input 1.

```
PRG 0                ;* Start of Program 0
HSPD=200             ;* Set high speed to 20000pps
ACC=500              ;* Set acceleration to 500RPM/sec
WHILE 1=1            ;* Forever loop
    X0                ;* Move to position 0
    X1000             ;* Move to position 1000
ENDWHILE             ;* Go back to WHILE statement
END                  ;* End Program 0

PRG 1                ;* Start of Program 1
WHILE 1=1            ;* Forever loop
    IF DI1=1          ;* If digital input 1 is on
        RGB=1
    ELSE
        RGB=2
    ENDIF            ;* End if statements
ENDWHILE             ;* Go back to WHILE statement
END                  ;* End Program 1
```

3. Serial Communicating Commands

3.1. Status Commands

Command	RW*	Description
CURQA	R	Motor current Q
CURDA	R	Motor current D
EX	RW	Encoder position
FLT	R	Fault condition. Bit 0 – Negative Limit Error Bit 1 – Positive Limit Error Bit 2 – Position Error Bit 3 – Current Error Bit 4 – Hall Sensor Error Bit 5 – Overcurrent Error Bit 6 – Over Voltage Error Bit 7 – Under Voltage Error Bit 8 – Overheat Error Bit 9 – Motor Connection Error Bit 10 – Emergency Switch On Bit 11 – Encoder Error
INPOSVAL	RW	In-position value.
MST	R	Returns current motor status. Bit 0 – Enabled Bit 1 – In Position Bit 2 – Moving Bit 3 – In Fault
PERR	R	Position error
POSD	R	Target position
VX	R	Velocity (estimated from encoder)

Table 3.0

3.2. LED Commands

Command	RW*	Description
LED	RW	Blue LED on the face of TITAN
RGB	RW	Internal RGB LED Bit 0 – Red LED Bit 1 – Green LED Bit 2 – Blue

Table 3.1

3.3. Motion Commands

Command	RW*	Description
ACC	RW	Target acceleration
ECLEARX	W*	Clear Fault
HMODE	RW	Homing Mode 0 – plus home 1 – minus home 2 – plus limit 3 – minus limit 4 – plus home + encoder index 5 – minus home + encoder index 6 – plus limit + encoder index 7 – minus limit + encoder index 8 – plus home using index channel 9 – minus home using index channel
HOMEX	W*	Perform Home
HSPD	RW	Target velocity
JOGXN	W*	Jog in Negative Direction
JOGXP	W*	Jog in Positive Direction
OLPHOLD	RW	Open Loop Position Hold (Valid for Stepper Only) Value 0 – Full time closed loop Value 1-100 – Percentage current hold at Enabled and Not Moving.
STOPX	W*	Stop Motion
SVOFF	W*	Disable Motor or Servo Off
SVON	W*	Enable Motor or Servo On
X	W	Perform Target position move

Table 3.2

3.4. Gain Commands

Command	RW*	Description
CGAINF	RW	Current Control Firmness. Range 0 to 100
IGAINF	RW	Integral Control Firmness. Range 0 to 100
PGAINF	RW	Position Control Firmness. Range 0 to 100
PRESETGAIN	RW	Sets all the Firmness values (PGAINF, VGAINF, and IGAINF) to the same value.
VGAINF	RW	Velocity Control Firmness. Range 0 to 100
DGENA	RW	Enable/disable Dynamic Gain
DGLGAIN	RW	Low speed Dynamic Gain
DGUGAIN	RW	High speed Dynamic Gain
DGLSPD	RW	Low speed used for the DGLGAIN
DGUSPD	RW	High speed used for the DGUGAIN
DGTYPE	RW	Dynamic Gain Type 0 – Profile velocity 1 – Actual velocity

Table 3.3

3.5. Standalone Program Commands

Command	RW*	Description
SAC	W	Standalone program control 1 – run all programs 2 – stop all programs 3 – pause all programs 4 – continue all programs 10 – store standalone programs and autorun to flash 40 – run program 0 41 – run program 1 42 – run program 2 43 – stop program 0 44 – stop program 1 45 – stop program 2 46 – pause program 0 47 – pause program 1 48 – pause program 2 49 – continue program 0 50 – continue program 1 51 – continue program 2
SASM[#]	R	Standalone program status of program 0, 1, or 2 0 – Idle 1 – running 2 – Paused 4 – Errored For program 0, use SASM0 command For program 1, use SASM1 command For program 2, use SASM2 command

Table 3.4

3.6. Limit Commands

Command	RW	Description
LHPOL	RW	Limit and Home input polarity and limit enable Bit 0 – Home Input Polarity Bit 1 – Limit Input Polarity Bit 2 – Enable/Disable Hard Limit
LIMPRO	RW	Sets action to perform when soft or hard limit is triggered 0 – Disable Motor 1 – Immediate Stop Motor with Servo On
SLIMNEG	RW	Negative Soft Limit Value
SLIMON	RW	Enable Soft Limit
SLIMPOS	RW	Positive Soft Limit Value

Table 3.5

3.7. Digital IO Commands

Command	RW*	Description
DIN	R	<p>Digital Inputs. TITAN-SVX has 8 digital inputs. Depending on whether TITAN is set in controller or pulse mode, some digital inputs are designated with special functions.</p> <p>DI1 – Pulse (Pulse Mode) DI2 – Dir (Pulse Mode) DI3 – Enable (Pulse Mode) DI4 – Clear Fault (Pulse Mode) DI5 – Reset Pos (Pulse Mode)/ Emergency Input (both) DI6 – Plus Limit (Controller Mode) DI7 – Home (Controller Mode) DI8 – Minus Limit (Controller Mode)</p>
DOUT	RW	<p>Digital Outputs. TITAN-SVX comes with 3 digital outputs. Depending on whether TITAN is set in controller or pulse mode, some digital outputs are designated with special functions.</p> <p>DO1 – Alarm (Pulse Mode) DO2 – Available for use DO3 – In-Pos (Pulse Mode)</p>

Table 3.6

3.8. Variable Commands

Command	RW	Description
VAN	RW	Set Variable Number to Read or Write
VAR	RW	Read Variable Value
VAW	RW	Write Variable Value

Table 3.7

3.9. Communication Commands

Command	RW	Description
TXDELAY	RW	Delay value to wait before a reply is sent back to the master.
IP_ADD	RW	IP address used for Ethernet communication
GATEWAY	RW	Gateway address used for Ethernet communication
SUBNET	RW	Subnet used for Ethernet communication
MACADD[0-5]	R	Assigned MAC address

Table 3.8

3.10. Fault Monitoring Commands

Command	RW	Description
CEMS	RW	Current Error Duration in Milliseconds
CEVAL	RW	Current Error Value in milliamps
ENAFc	RW	Enable Position and Current Fault Monitoring Bit 0 – Enable Position Error Monitoring Bit 1 – Enable Current Error Monitoring
ESTOP	RW	Digital Input 5 configured as an emergency input. When this input is triggered, the motor is disabled and fault flag is set. Value 0 – disable emergency input Value 1 – Enable emergency active high Value 2 – Enable emergency active low
OCUR	RW	Peak Motor Current Monitoring Value. The default value of 8A is always set at power up. Peak current monitoring is always enabled for the protection of the motor and driver. OCUR value can be changed to a lower current threshold value. OCCUR cannot be raised above 8A .
OVOL	RW	Peak Power Supply Voltage Monitoring Value. The default value of 65V is always set at power up. Peak voltage monitoring is always enabled for the protection of the motor and driver. OVOL value can be changed to set to a lower voltage threshold value. OVOL cannot be raised above 65V .
PEMS	RW	Position Error duration Milliseconds
PEVAL	RW	Position Error Value in encoder counts
UHEA	RW	Overheat Temperature Monitoring Value. The default value of 80 deg C is always set at power up. Temperature monitoring is always enabled for the protection of the motor and driver. UHEA can be lowered but not be raised higher than 80.
UVOL	RW	Under Power Supply Voltage Monitoring Value. The default value of 10V is always set at power up. Under-voltage monitoring is always enabled for the protection of the motor and driver. UVOL value can be changed to set to a higher voltage threshold value. UVOL cannot be lower than 10V or raised above 65V .

Table 3.9

3.11. Miscellaneous Commands

Command	RW	Description
FIRMVS	R	Returns Firmware Version
MOTNAME	RW	Returns Motor Product Name
NETID	RW	Returns and sets RS485 Network ID. The range is 1 to 99. Once Network ID is set, it must be stored to flash and controller reboot for the new ID to take effect.
PWRC	R	Power source current usage
PWRV	R	Power source voltage
RESET	W	Resets the controller.
STORE	W	Stores all parameters to flash memory. Once the store command is issued, wait at least 500msec before communication started.
TEMP	R	FET temperature
SYSTIME	RW	Returns number of seconds since the last power up.
RUNTIME	RW	Returns number of seconds motor is enabled and moving (velocity non-zero). Value is reset at zero at power up.

Table 3.10

3.12. Trend Monitoring Commands

Command	RW	Description
TRENDSEL	RW	Select which trend to monitor 1 – Power 2 – Reserved 3 – Temperature
TRENDSEC	RW	Trend Second Value Array Range 0 to 59, Example TRENDSEC[0]
TRENDMIN	RW	Trend Value Minute Resolution. Array Range 0 to 59, Example TRENDMIN[12]
TRENDHOUT	RW	Trend Value Hour Resolution. Array Range 0 to 23, Example TRENDHOUR[8]
TRENDDAY	RW	Trend Value Day Resolution. Array Range 0 to 29, Example TRENDDAY[16]
SYSTIME	RW	Returns time in seconds since the power up
RUNTIME	RW	Returns time in seconds with Motor Enabled and Moving
FLTTIME	RW	Fault Trigger Time

Table 3.11

3.13. Force Control Commands

Command	RW	Description
FCVA	RW	Velocity used to move to the start search position (FCPB)
FCAA	RW	Acceleration to the start search position (FCPB)
FCVB	RW	Velocity used to search for the trigger detected position (FCPT)
FCAB	RW	Acceleration used to search for the triggered detected position (FCPT)
FCVC	RW	Velocity used to move back to initial position (FCPA)
FCAC	RW	Acceleration used to move back to initial position (FCPA)
FCPA	RW	Initial position
FCPB	RW	Start deceleration position to detect the trigger
FCPT	RW	Trigger detected position
FCDP	RW	Delta position to apply continuous force
FCPM	RW	Maximum position to move to find the trigger
FCDL	RW	Delta position to move out from the continuous force
FCCA	RW	Current used for moving to start detect position
FCCB	RW	Current used when moving to detect the trigger position
FCCC	RW	Current used to determine the trigger position
FCCF	RW	Current used after finding trigger position for continuous force
FCTP	RW	Time duration of the constant force
FCMODE	RW	Force control mode
FCSTAT	R	Force control status
FCSTEP	RW	Force control step
FCENA	RW	Enable force control
FCCYC	RW	Number of cycles to run the force control routine
FCCMD	W	Force control command

Table 3.12

4. Software Overview

The TITAN-SVX has Windows compatible software that allows for USB, RS485, or Ethernet communication. Standalone programming, along with all other available features of the TITAN-SVX, will be accessible through the software. It can be downloaded from the Arcus Servo website.

4.1. Software Installation

Download the latest version of the TITAN-SVX software from the Arcus Servo website (www.arcusservo.com) and run the installation steps

If you have previous version of the TITAN-SVX software already installed, the installation program will prompt to uninstall the previous version. Please uninstall the previous version so that the latest version will be installed.

The first installation window is shown in figure 4.0.

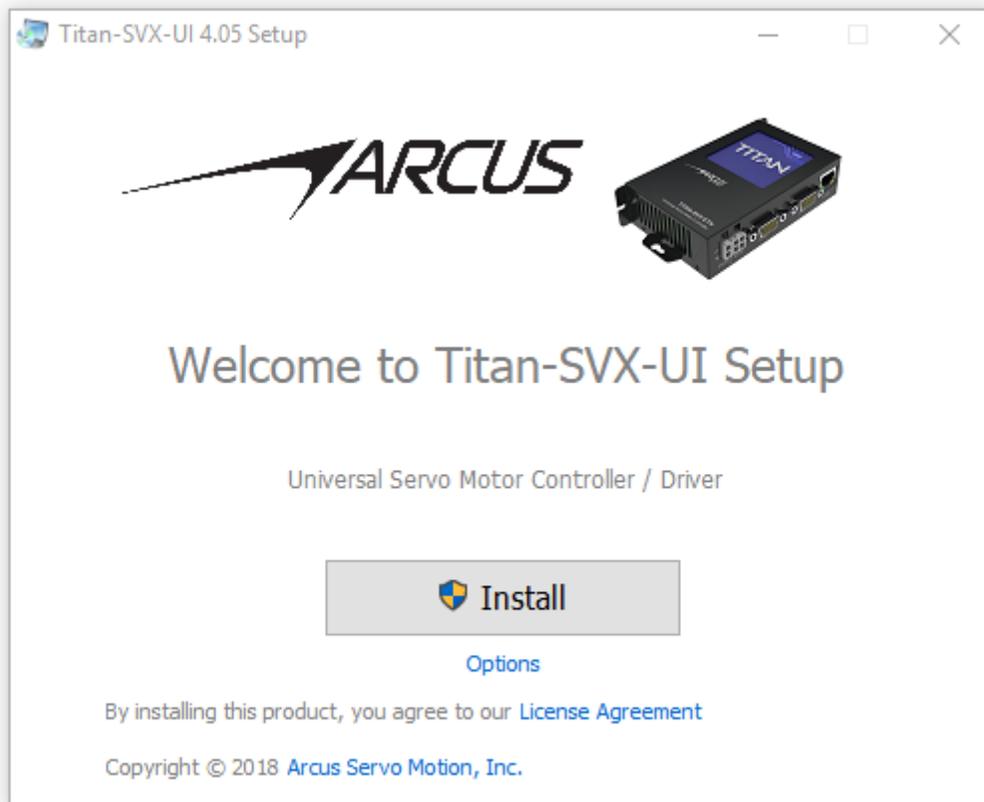


Figure 4.0

The License Agreement can be viewed by clicking the corresponding link. Please review and confirm the conditions listed here.

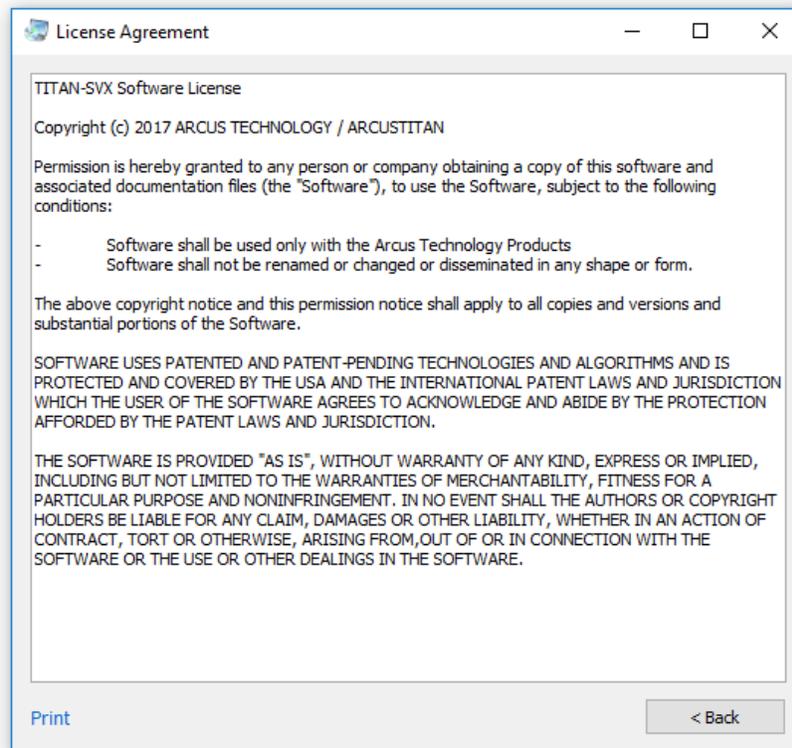


Figure 4.1

The Options link will display the installation options shown in figure 4.2. The options can be adjusted here if desired.

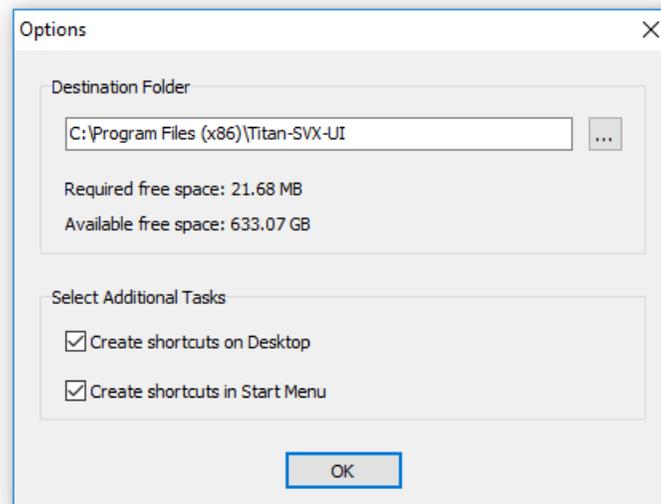


Figure 4.2

Once the options are confirmed, press the Install button from the main install screen.

Note that you may get an Account Control Installation Notice to install the TITAN-SVX executable program. Please select “Yes” and proceed with the installation. Once installation is complete, following screen will show to indicate a successful installation.

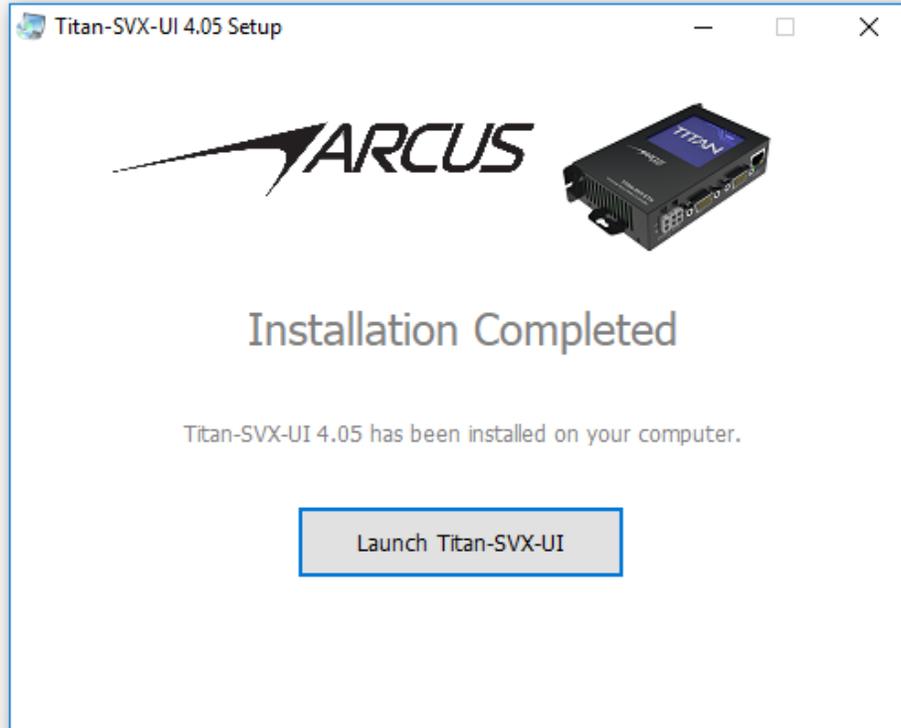


Figure 4.3

4.1.1. TITAN Directory

TITAN-SVX software uses a dedicated directory to keep the various application files. The main TITAN-SVX executable software will be installed in the following directory by default:

C:\Program Files (x86)\Titan-SVX-UI

The other directory TITAN-SVX uses is in the following location:

C:\Users\Public\Documents\TITAN-SVX-V3

Under this dedicated directory, following sub directories are used for various files for the TITAN-SVX software.

- \EXE** – sub directory that contains the stand-alone compiler files
- \MotorDB** – motor database files
- \SystemDB** – system database files
- \MotionPrograms** – standalone motion program files

Additional directories may be created and used for new features and enhancements.

4.2. Communication and Setup Requirements

TITAN-SVX software can use a serial communication port to communicate with the TITAN-SVX.

For most systems a USB to RS485 communication converter will be needed, such as the ACE-USB-485 from Arcus Technology.



Figure 4.4

Other RS485 to USB converters can be used as well.

Connect the 485+ and 485- (and GND if necessary) between the USB/485 converter and the TITAN-SVX.

Following is an example of a connection between a PC and TITAN-SVX.

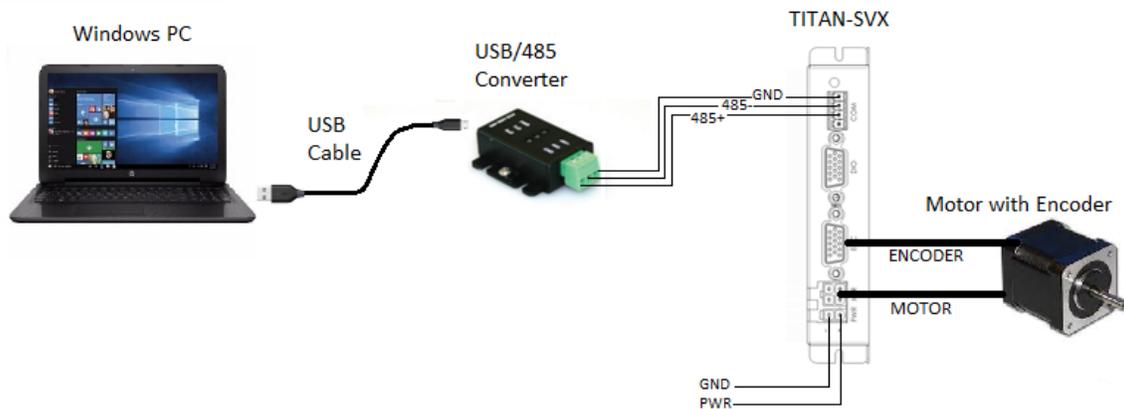


Figure 4.5

4.3. Starting the TITAN-SVX UI Software

The main screen of the TITAN-SVX GUI is shown in figure 4.6. This screen will allow you navigate back and forth between various configuration and control features of the TITAN-SVX.

Important Note: TITAN-SVX software communicates with a single TITAN-SVX controller at a time.



Figure 4.6

4.4. Communication

The Communication screen can be accessed by clicking the Communication button. This screen will allow you to open communication to a TITAN-SVX, search the network for existing TITAN-SVX, and update communication settings.



Figure 4.7

4.4.1. USB/Serial Communication

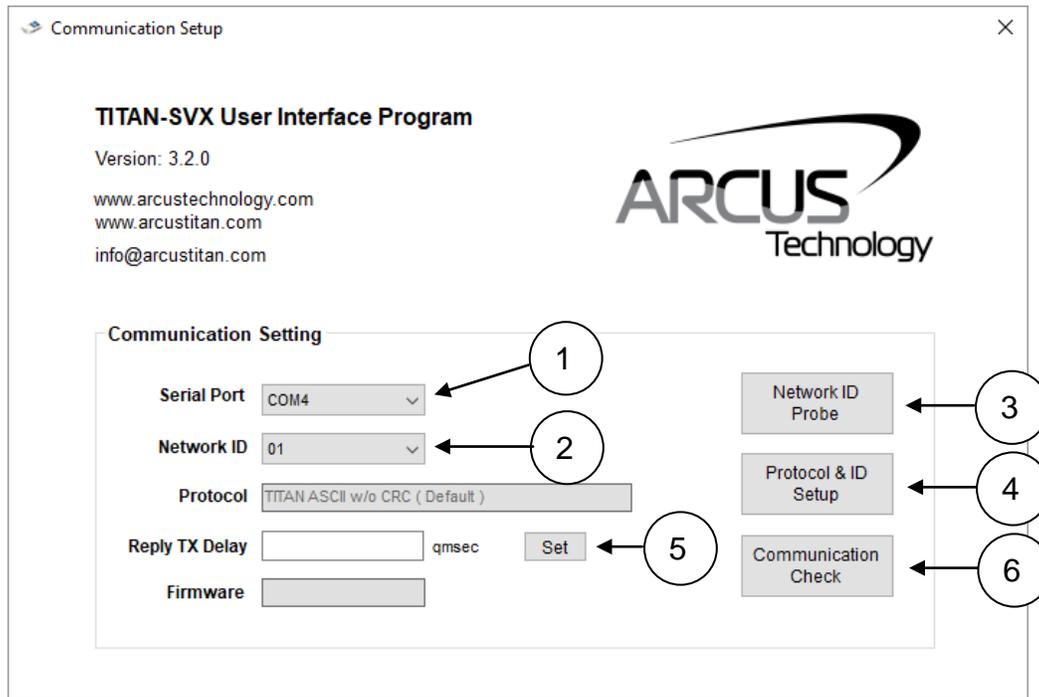


Figure 4.8

1. **Serial Port:** The Serial Port dropdown list will list all active communication ports that are available on the PC. In order to communicate with the TITAN-SVX, be sure to select to the correct COM port. If you do not see any communication ports on the list, check to make sure the drivers for your COM port device are installed correctly.
2. **Network ID:** The default Network ID is 1. If a different network ID has been setup on the TITAN-SVX, the appropriate ID can be selected.
3. **Network ID Probe:** You can see the list of all TITAN-SVX that are connected to the RS485 network by clicking on the Network ID Probe button. The software will probe network ID's 1 to 24 and list any TITAN-SVX that are found.
4. **Protocol & ID Setup:** TITAN-SVX supports TITAN-ASCII and MODBUS protocols. To see the protocol list shown in figure 4.9, click the Protocol & ID Setup button.

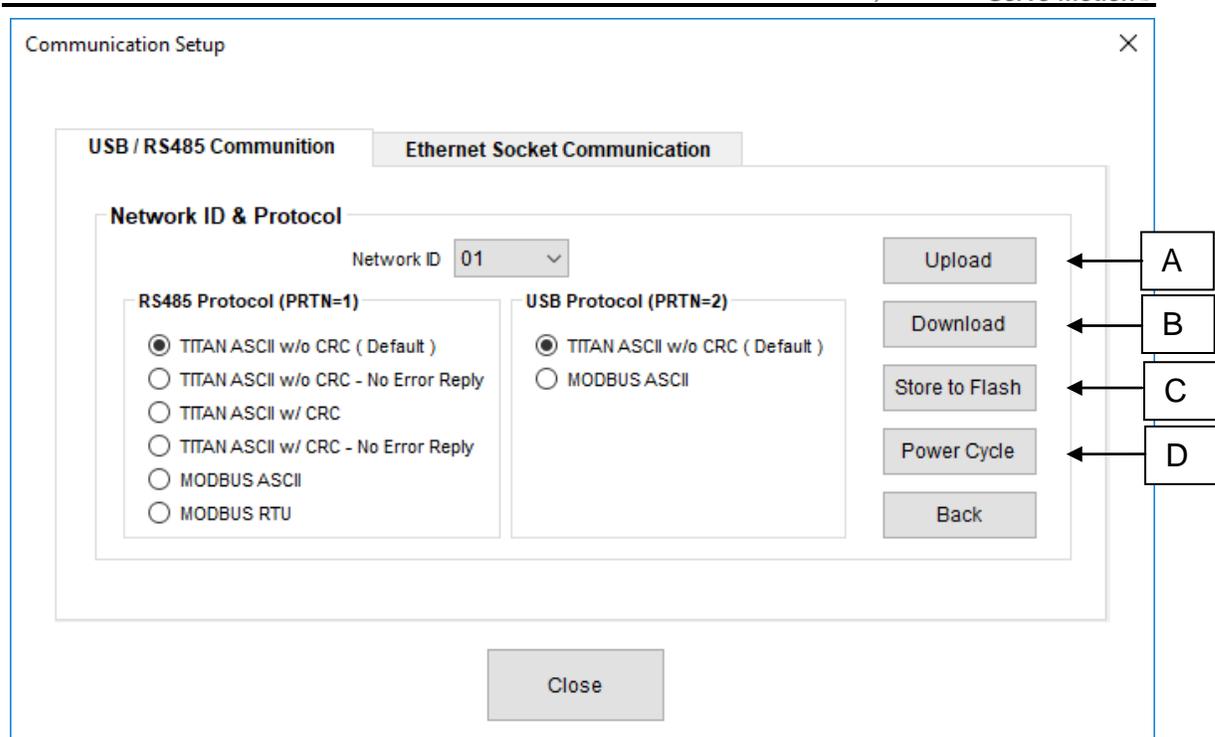


Figure 4.9

For detailed information of the different protocols that are available on the TITAN-SVX, please refer to the TITAN Communication Manual. *The following buttons use Special Request Commands that are common to all available protocols on the TITAN-SVX.*

- A. *Upload*: Upload the communication settings from the TITAN controller.
- B. *Download*: Download the selected Network ID and the Communication Protocol settings. The changes will take effect immediately.
- C. *Store*: Store the settings to the flash. Note that store to flash will save ALL parameters, including any downloaded communication settings, to the flash.
- D. *Power Cycle*: Power cycle will perform a soft power cycle of the controller.

Important Protocol Notes:

- **TITAN-SVX software uses TITAN ASCII protocol without CRC to communicate with the TITAN controller. If any other protocol is selected, TITAN software will NOT be able to communicate with the TITAN-SVX controller.**
- **Limited Special Request Commands are supported on all the communication protocols. Special Request Commands can be used to upload and download the communication protocol and the network ID, to perform flash store, and to power cycle. For example, from MODBUS-RTU protocol, TITAN-SVX controller can**

be changed back to TITAN ASCII protocol easily using the Special Request Commands. Please refer to the TITAN Communication Manual for details of the Special Request Commands.

- **Special Request Commands use a special broadcast type of command and only one TITAN-SVX unit is to be connected on the RS485 network. If multiple TITAN-SVX units are connected on the RS485, Special Request Command will not work properly.**

5. *Reply TX Delay:* The TX Delay parameter delays the reply message from the TITAN-SVX to the RS485 master. When communicating on the RS485 line, only one device (Master or Slave) can use the RS485 line. When multiple units hold the RS485 line or try to send messages at the same time, communication collision will occur, which will result in corrupted messages. In the event that the Master sends a command and does not release the RS485 line fast enough before the TITAN-SVX sends the reply, the TX delay parameter can delay the response from the TITAN-SVX to avoid RS485 conflicts.

TX Delay is in quarter milliseconds resolution. For example, a TX Delay value of 4 will result in a 1 millisecond delay before the reply is sent out.

6. *Communication Check:* Once the Serial Port and Network ID are selected, click the Communication Check Button to establish communication.

TX Delay value and the TITAN-SVX firmware version will also be uploaded and shown when successful communication is established.

4.4.2. Ethernet Communication

The Ethernet communication section is only applicable to the TITAN-SVX-ETH model.

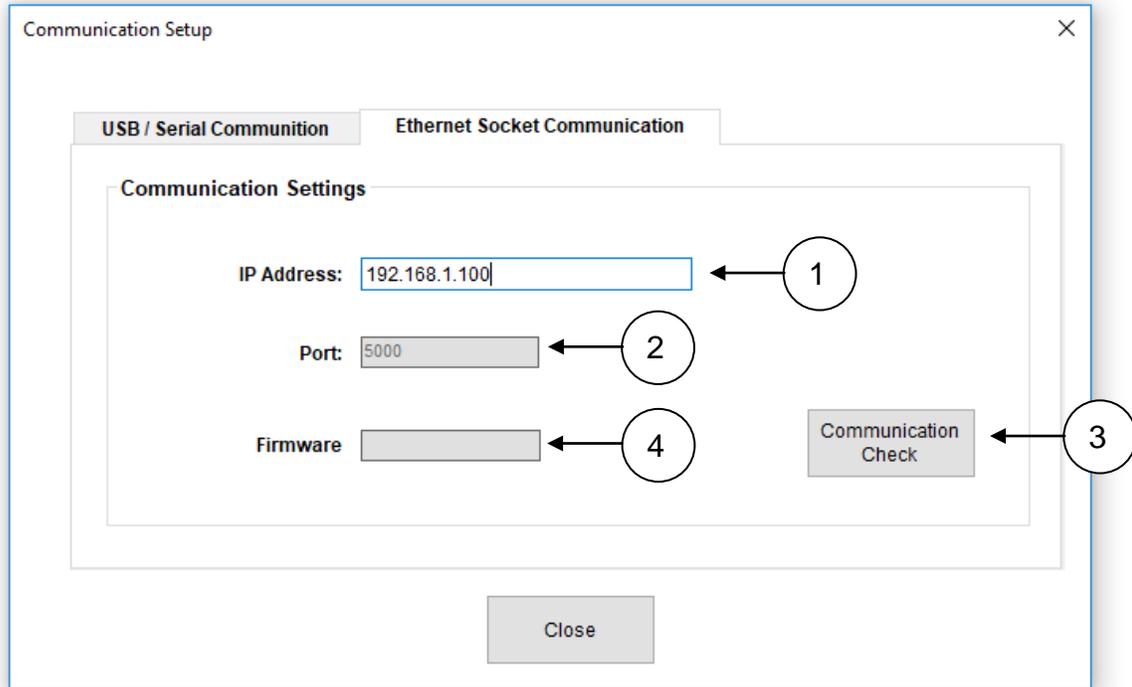


Figure 4.10

1. *IP Address*: Enter the IP address of the TITAN-SVX-ETH. The factory default IP address is 192.168.1.100.
2. *Port*: The port of the TITAN-SVX-ETH is fixed to 5000
3. *Check Communication*: Once the IP address is entered, click the Communication Check Button to establish communication.
4. *Firmware*: Once a successful connection is made, the firmware version will be displayed.

4.5. Motor



Figure 4.11

Motor Parameters ✕

Motor Information

Motor Name: <input type="text" value="STPM-23-2-R16000"/>	Rated Voltage: <input type="text" value="24"/> V
Motor Mech: <input type="text" value="Rotary"/>	Max Current: <input type="text" value="3"/> A
Motor Type: <input type="text" value="Stepper (2 Ph)"/>	Pole Pairs: <input type="text" value="50"/>
Motor Direction: <input type="text" value="Normal"/>	Magnet Pitch (Linear Motor Only): <input type="text" value="--"/> mm

Position Sensor Parameters

Encoder Type: <input type="text" value="Encoder with Index"/>	Encoder Resolution: <input type="text" value="16000"/> cnt/pitch
Hall Sensor Type: <input type="text" value=""/>	Encoder Direction: <input type="text" value="Invert"/>

Motor Electrical Parameters

EMF Constant (Ke): <input type="text" value="18.2086"/> mV/(rpm)
Peak Resistance: <input type="text" value="5.86032"/> Ohm
d-axis Inductance: <input type="text" value="11.3986"/> mH
q-axis Inductance: <input type="text" value="12.766"/> mH
Electric Bias Angle: <input type="text" value="-131"/> deg

Motor Mechanical Parameters

Maximum Speed: <input type="text" value="3000"/> rpm
Enable FWC <input checked="" type="checkbox"/>
Motor Inertia: <input type="text" value="0.0184266"/> g.m2
Viscous Friction: <input type="text" value="0.00284781"/> mNm/rpm
Coulomb Friction: <input type="text" value="8.45474"/> mNm

Open MDB

Save

Upload

Download

Store Settings

MDB Wizard

Unlock for Edit

Figure 4.12

TITAN-SVX Operation Manual

page 45

Rev 4.05

4.5.1. Motor Information

Motor Information

Motor Name: <input style="width: 80%;" type="text" value="TM-STP-23-2-E100"/>	Rated Voltage: <input style="width: 40%;" type="text" value="24"/> V
Motor Mech: <input style="width: 80%;" type="text" value="Rotary"/>	Max Current: <input style="width: 40%;" type="text" value="3"/> A
Motor Type: <input style="width: 80%;" type="text" value="Stepper (2 Ph)"/>	Magnet Pitch (Linear Motor Only) <input style="width: 40%;" type="text" value="--"/> mm
Motor Direction: <input style="width: 80%;" type="text" value="Normal"/>	

Figure 4.13

1. *Motor Name*: A maximum of 32 characters can be used to name the motor.
2. *Motor Mechanism*: Select whether the motor is rotary or linear.
3. *Motor Type*: Select the motor type. Stepper, Brushless PMSM, or DC Voice Coil options are available
4. *Motor Direction*: Normal or Inverted motor polarity.
5. *Rated Voltage*: 24VDC to 48VDC voltage of power supply that is used to power the TITAN-SVX
6. *Max Current*: Set the maximum allowable current from the TITAN-SVX to the motor. The motor specification should be referenced for value. The TITAN-SVX allows a maximum of 8A.
7. *Magnet Pitch*: This setting is used only for linear motors
 - a. For brushless linear PMSM motors. Describes the magnet pair to magnet pair pitch (N/S to N/S pitch)
 - b. For Voice Coil motor, maximum stroke length matching the encoder resolution. For example, if the stroke of the voice coil is 20mm and the encoder resolution is 1 micron, magnet pitch will be 20 and the encoder resolution will be 20,000.

4.5.2. Position Sensor Parameters

Position Sensor Parameters

Encoder Type: <input style="width: 80%;" type="text" value="Encoder with Index"/>	Encoder Resolution: <input style="width: 40%;" type="text" value="16000"/> cnt/pitch
Hall Sensor Type: <input style="width: 80%;" type="text"/>	Encoder Direction: <input style="width: 40%;" type="text" value="Invert"/>

Figure 4.14

1. *Encoder Type*: Incremental encoders are supported by the TITAN-SVX
2. *Hall Sensor Type*: Only applicable for Brushless motors with Hall Sensors.
3. *Encoder Resolution*: For Rotary motors, the units is encoder edges/revolution and for linear motors, the units is encoder edges/magnet pitch.
4. *Encoder Direction*: Inverted or Normal direction of the encoder counting. Note that encoder direction can also be changed by switching the Encoder A and B wiring.

4.5.3. Motor Electric Parameters

Motor Electrical Parameters

EMF Constant (Ke): V/(rpm)

Resistance List: Ohm

d-axis Inductance: mH

q-axis Inductance: mH

Electric Bias Angle: deg

Figure 4.15

1. *Motor Constant*: also known as motor electric constant
2. *Motor Resistance*: resistance of the winding
3. *Motor Inductance*: q and d axis inductance of the motor
4. *Motor Bias Angle*: the bias angle of the motor

4.5.4. Motor Mechanical Parameters

Motor Mechanical Parameters

Maximum Speed: rpm

Enable FWC

Motor Inertia: g.m²

Viscous Friction: mNm/rpm

Coulomb Friction: mNm

Figure 4.16

1. *Maximum Speed*: The maximum allowable speed the TITAN-SVX will be able to move the connected motor.
 - a. *Enable FWC*: enable/disable the field weakening condition. If enabled the motor will be able to move at faster speeds with reduced torque output.
2. *Motor Inertia*: The built in inertia of the motor.
3. *Viscous & Coulomb Friction*: Friction parameters of the motor.

4.5.5. Motor Database Wizard

TITAN-SVX is capable of controlling various types of commonly used motors:

- Rotary 2-Phase Stepper Motors
- Rotary 3-Phase Brushless PMSM Motors
- Linear 3-Phase Brushless PMSMS Motors
- Linear Voice Coil DC Motors

Arcus Technology supplies various TITAN motors that already come with the motor database file ready to be used with the TITAN-SVX. For these motors, Motor Database Wizard is not needed since the motor database files are already generated and available to use with the TITAN-SVX.

For motors that are not supplied from Arcus Technology, Motor Database Wizard can be used to generate the motor database file. The following steps will create a motor database file.

Note: For Rotary Motor Auto Detect, make sure that the motor is free and not connected to any mechanism.

Step 1 - Motor Database Setup Acknowledgement

Creating the motor database will involve movement of the motors and will require extreme caution. Confirm with acknowledgement before proceeding with the setup.

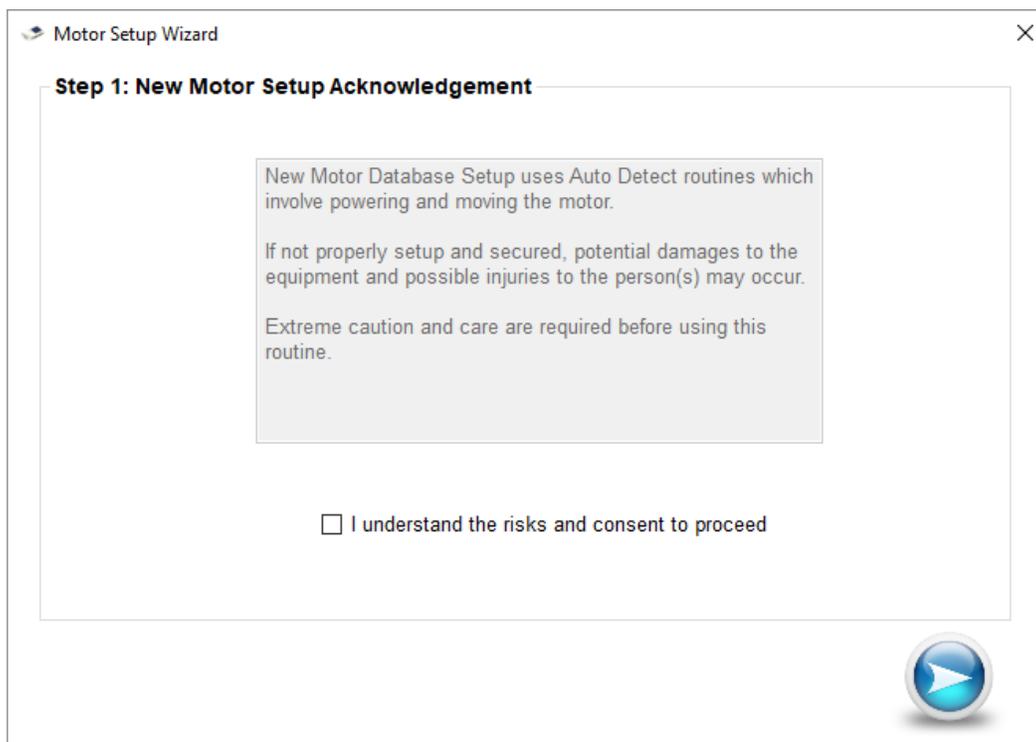


Figure 4.17

Step 2A – Motor Type Selection (Stepper Motor)

Once the TITAN-SVX reset is complete, you will see following screen. The stepper motor option is selected by default.

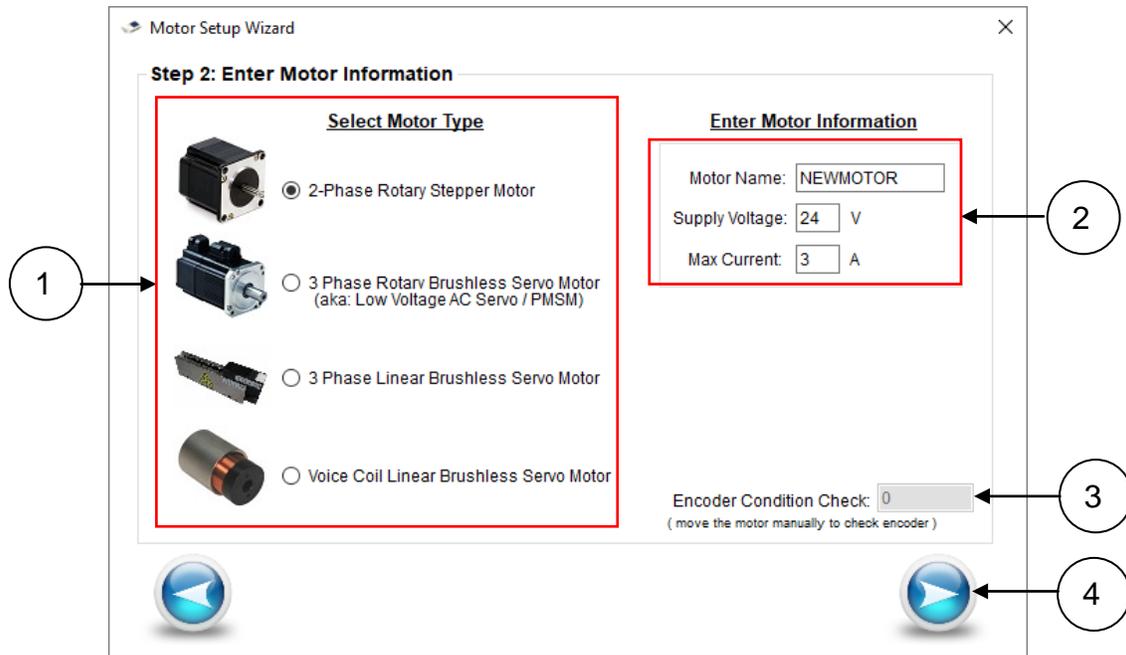


Figure 4.18

1. *Select Motor Type*: Select the motor type connected to the TITAN-SVX
2. *Motor Information*: Type the motor name, supply voltage, and maximum current setting for the motor.
 - TITAN-SVX supports from 24VDC to 48VDC.
 - TITAN-SVX supports up to maximum of 8A.
3. *Encoder Condition Check*: You will be able to see the encoder counter value. Note that TITAN-SVX supports only incremental encoders.
4. Once the proper motor type is selected and motor information is entered, click the next screen button.

Step 2B – Motor Type Selection (3 Phase Motor)

For a 3 Phase Rotary motor, you will see following screen:

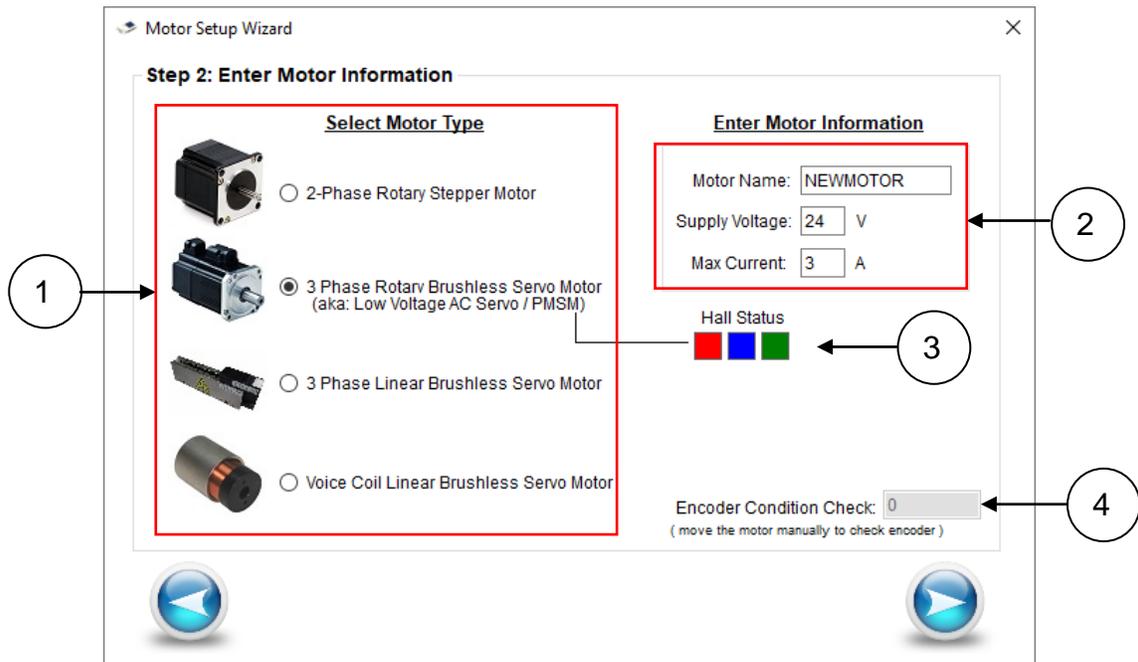


Figure 4.19

1. *Select Motor Type*: Select the motor type connected to the TITAN-SVX
2. *Motor Information*: Type the motor name, supply voltage, and maximum current setting for the motor.
 - TITAN-SVX supports from 24VDC to 48VDC.
 - TITAN-SVX supports up to maximum of 8A.
3. *Hall Status*: For a 3-Phase Rotary Motor, the hall sensor status will be displayed. If the hall sensor is connected, the hall sensor status will change as the shaft is rotated. If the hall status is all on or all off, this will indicate that no hall sensor is detected
4. *Encoder Condition Check*: You will be able to see the encoder counter value. Note that TITAN-SVX supports only incremental encoders.

Step 2C – Motor Type Selection (3 Phase Linear Motor)

For a 3 Phase Linear Motor, you will see the following screen:

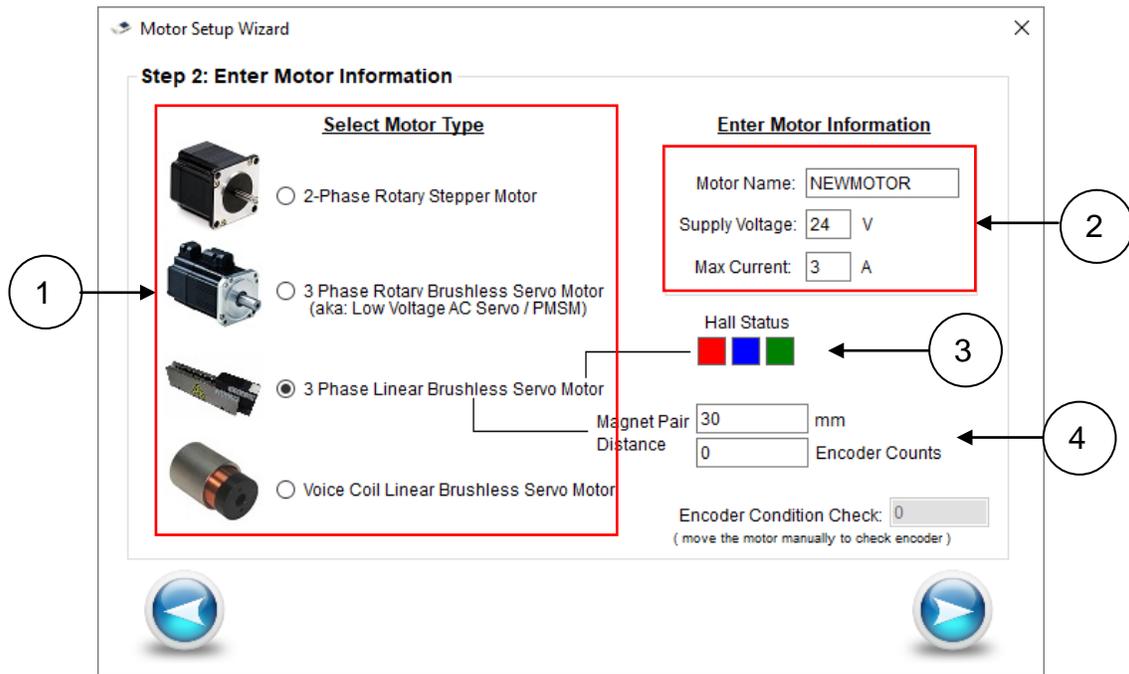


Figure 4.20

1. *Select Motor Type*: Select the motor type connected to the TITAN-SVX.
2. *Motor Information*: Type the motor name, supply voltage, and maximum current setting for the motor.
 - TITAN-SVX supports from 24VDC to 48VDC.
 - TITAN-SVX supports up to maximum of 8A.
3. *Hall Status*: For a 3-Phase Rotary Motor, the hall sensor status will be displayed. If the hall sensor is connected, the hall sensor status will change as the shaft is rotated. If the hall status is all on or all off, this will indicate that no hall sensor is detected
4. *Magnet Pair/Encoder Counts*: For 3-Phase Brushless Linear Motor, the linear magnet pitch and resolution is required. The following diagram shows the magnet pitch distance. Enter this value in mm.

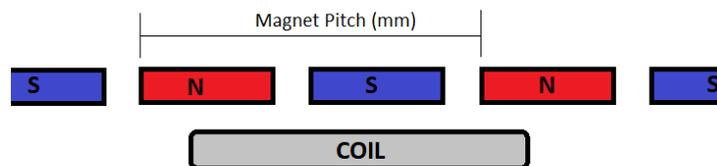


Figure 4.21

The magnet pitch encoder count value is the number of encoder counts N-N magnetic pitch. For example, if the magnet pitch distance is 20mm and the encoder resolution is 1 micron, encoder count value for the magnet pitch is 20,000 counts.

Step 2D – Motor Type Selection (Voice Coil)

For Voice Coil DC motor selection, following screen will show:

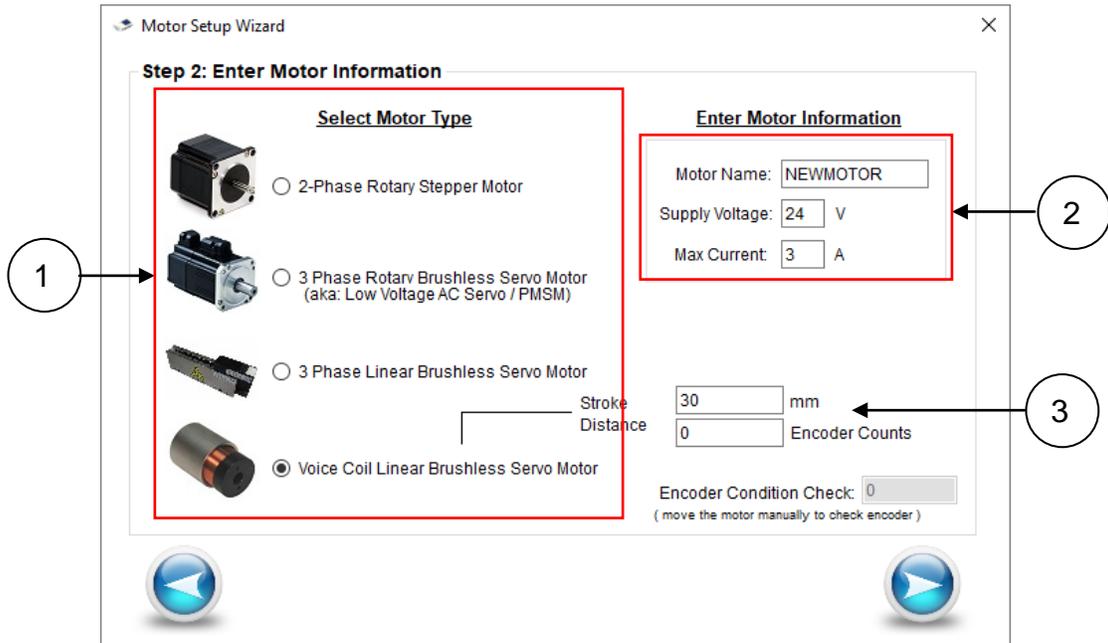
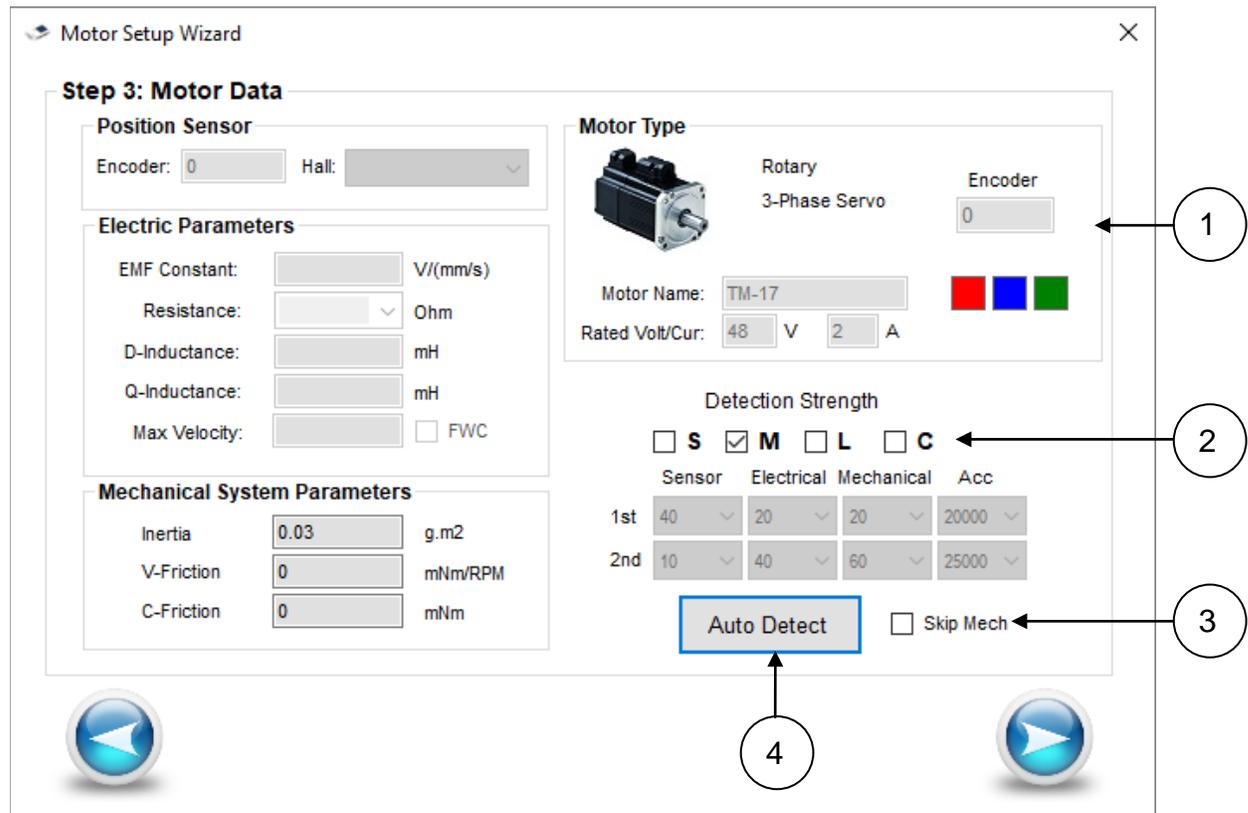


Figure 4.22

1. *Select Motor Type*: Select the motor type connected to the TITAN-SVX
2. *Motor Information*: Type the motor name, supply voltage, and maximum current setting for the motor.
 - TITAN-SVX supports from 24VDC to 48VDC.
 - TITAN-SVX supports up to maximum of 8A.
3. *Stroke Distance/Encoder Counts*: Enter the voice coil stroke length and the encoder counts for the defined stroke length. For example, if the voice coil stroke length is 25mm and 1 micron linear encoder is used, enter 25 for the stroke distance and 25,000 for the encoder counts.

Step 3 – Motor Data

The next step in Motor Database Setup will be the following screen.



Motor Setup Wizard

Step 3: Motor Data

Position Sensor
Encoder: 0 Hall:

Electric Parameters
EMF Constant: V/(mm/s)
Resistance: Ohm
D-Inductance: mH
Q-Inductance: mH
Max Velocity: FWC

Mechanical System Parameters
Inertia: 0.03 g.m2
V-Friction: 0 mNm/RPM
C-Friction: 0 mNm

Motor Type
Rotary 3-Phase Servo Encoder: 0
Motor Name: TM-17
Rated Volt/Cur: 48 V 2 A

Detection Strength
 S M L C

	Sensor	Electrical	Mechanical	Acc
1st	40	20	20	20000
2nd	10	40	60	25000

Auto Detect Skip Mech

Figure 4.23

1. *Motor Type*: Data entered in the previous screen is displayed again here. Confirm the information before proceeding.
2. *Detection Strength*: Select the detection strength that will be used in the motor parameter detection routine.
 - S – Select for small size motors such as small NEMA 11 size motors.
 - M – Select for NEMA 17 and 23 size motors.
 - L – Select for NEMA 34 size motors
 - C – Select for special motors with customized detection strength values
3. ***Skip Mechanical Detection is for linear motors only.*** For rotary motors, mechanical detection must be done. The parameters will be required in order to perform the *system* mechanical parameter detection.
4. Auto Detect will start the auto detection routine. This will involve moving the motors. Extreme caution should be taken.

Note: For Rotary Motor Auto Detect, make sure that the motor is free and not connected to any mechanism.

Following screen will show when Auto Detect is performed.

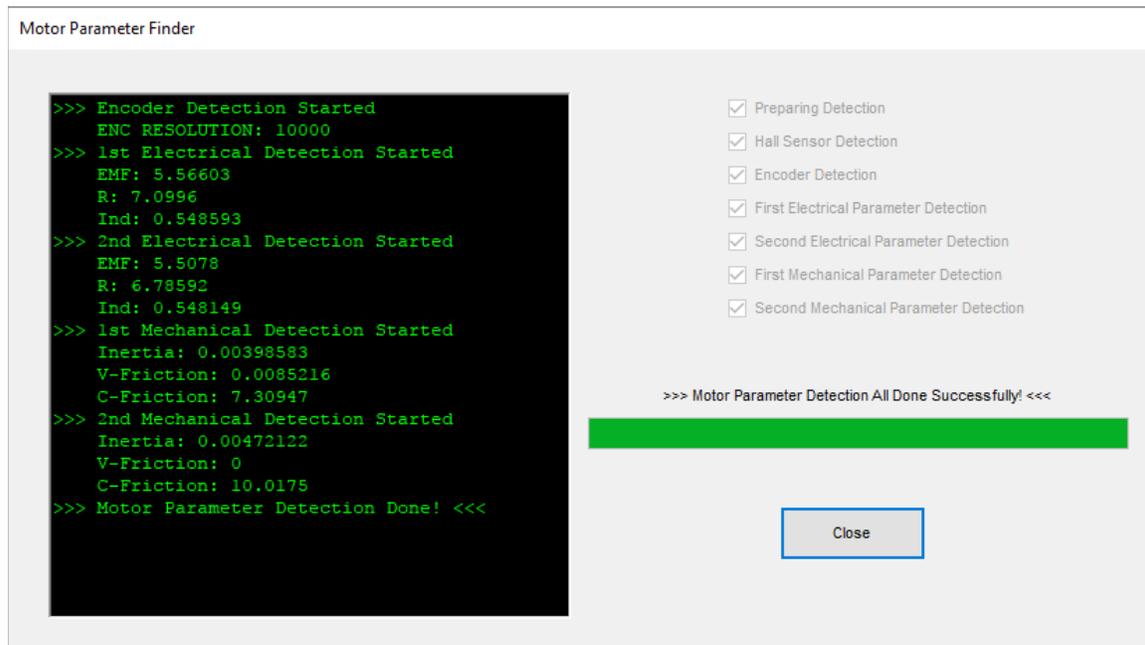


Figure 4.24

The steps below will be performed in the motor parameter auto detect routine. A message will appear after each step. Confirmation is needed to proceed. This will insure no damage is done to the motor/system as the auto detection routine is performed.

1. Hall Sensor Type Detection (if applicable)
2. Motor Encoder Resolution Detection
3. Motor Electrical Parameter Detection. This detection process will be done twice and the second set of parameters will be used. This routine will detect the following parameters:
 - a. Motor EMF constant
 - b. Motor Resistance
 - c. Motor Inductance
4. Motor Mechanical Parameter Detection (if applicable). This detection process will be done twice and the second set of parameters will be used. This routine will detect the following parameters:
 - a. Motor Inertia
 - b. Viscous Friction
 - c. Coulomb Friction

Once Motor Parameter Detection is done successfully, the found motor parameters will be automatically downloaded to the TITAN-SVX. Proceed to the next step of the Motor Database Wizard.

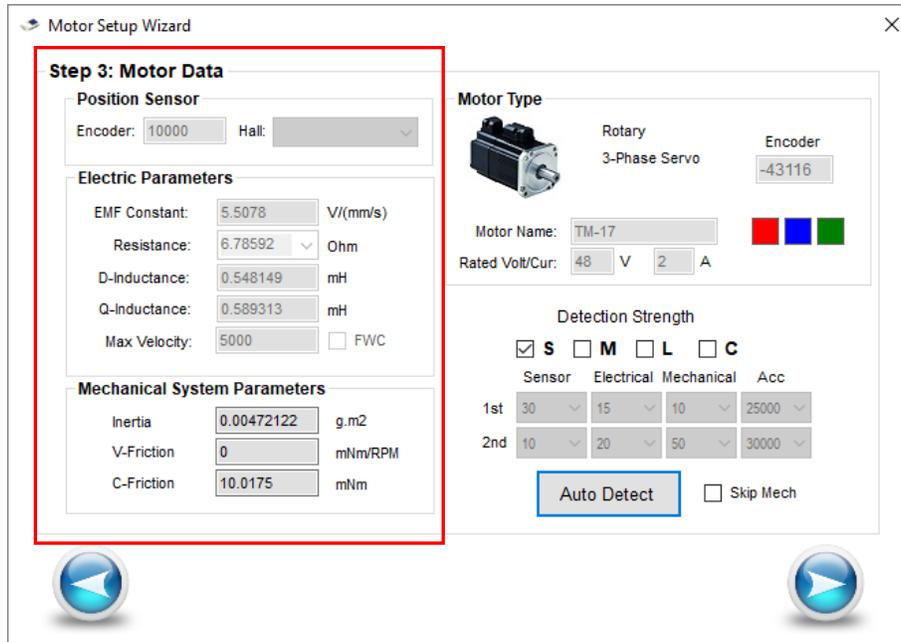


Figure 4.25

Step 4 – Motor Data Summary

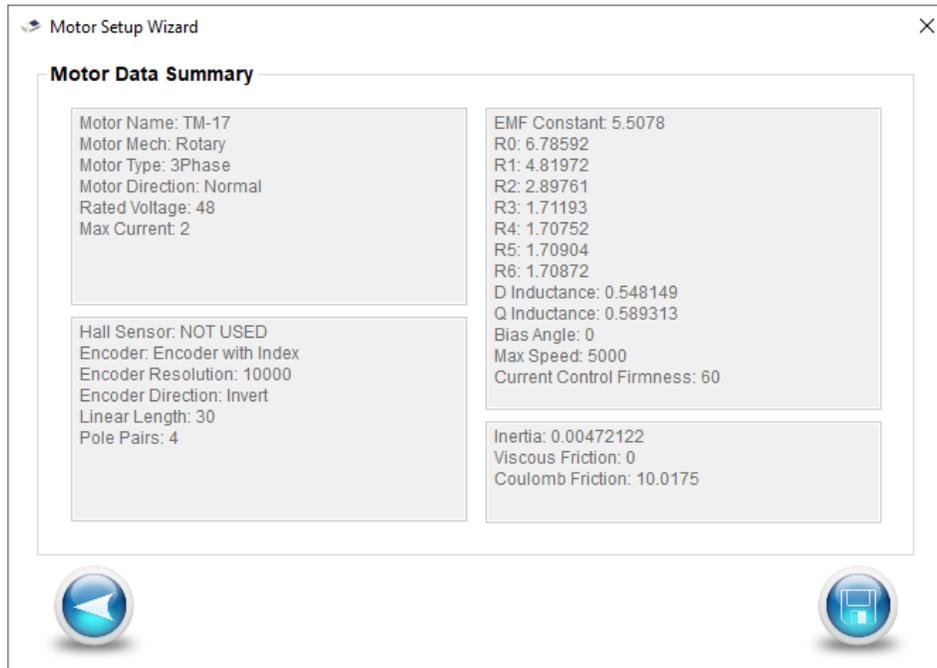


Figure 4.26

All the parameters found during the Motor Database Wizard are displayed for review.

Save the motor parameters to the database file using the  Save button. Note that the motor database file will be saved in the following directory by default.

C:\Users\Public\Documents\TITAN-SVX-V3\MotorDB

If a save of motor database file is not desired, the Motor Data Summary dialog box can be closed.

4.6. Tuning



Figure 4.27

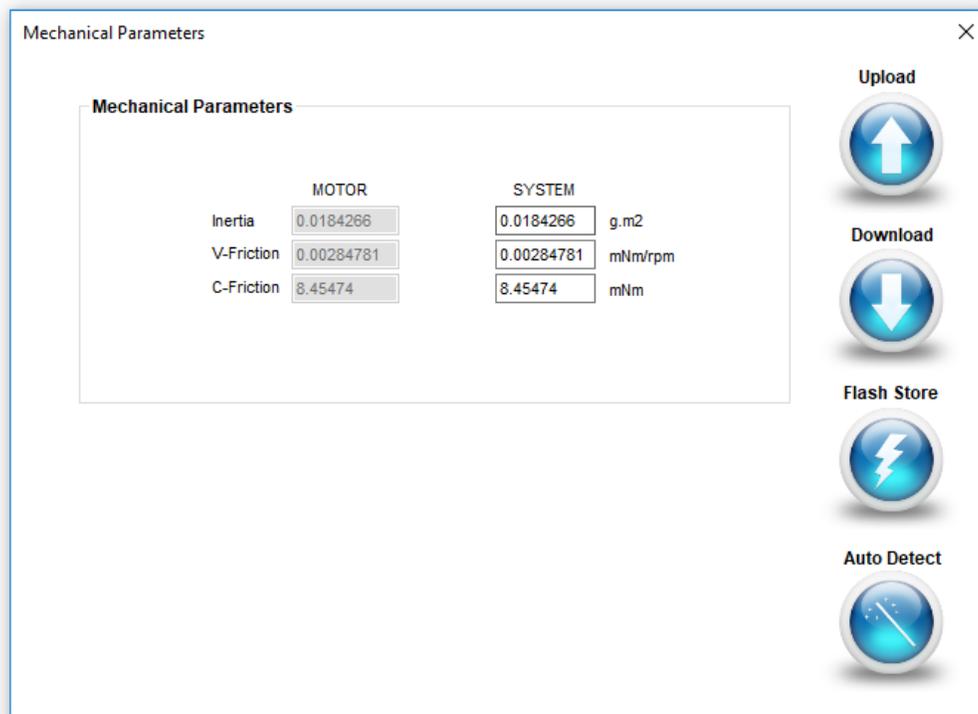


Figure 4.28

The TITAN-SVX uses the following three parameters in the closed loop control algorithm.

1. *Inertia*: System Inertia is also known as the mass of the system being controlled. This includes the motor inertia and the mechanical system inertia. For example, if the motor is connected to the belt drive linear axis that moves a 1kg mass, system inertia value include the motor inertia, belt and pulley inertia, bearing carriage inertia, and the 1kg mass inertia.
2. *Viscous Friction*: Viscous friction value is independent of the inertia and is the proportional increase in the friction value with the velocity of the motion.
3. *Coulomb Friction*: Coulomb friction is the initial friction that is present in the system but remains constant regardless of the velocity.

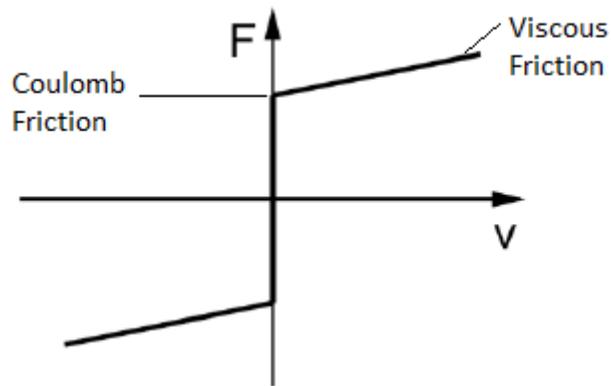


Figure 4.29

Getting the three parameters as accurately as possible will result in better closed loop control of the system.

4.6.1. Mechanical Parameters

Mechanical Parameters			
	MOTOR	SYSTEM	
Inertia	0.0175004	0.0175004	g.m2
V-Friction	0.0118531	0.0118531	mNm/rpm
C-Friction	6.23173	6.23173	mNm

Figure 4.30

1. *Motor Mechanical Parameters*: three motor mechanical parameters are shown. For rotary motors, these motor mechanical parameters should have been detected in the motor setup. For linear motors, these parameters will be at default values.
2. *System Mechanical Parameters*: three system mechanical parameters. These values should be equal or higher than the motor parameters, especially the inertia value. Viscous and Coulomb friction values may be lower than the motor friction values.

4.6.2. Mechanical Parameter Auto Detect Wizard

Mechanical system parameter detection routine will require movement of the motor that may cause sudden and erratic motion. Extreme caution and care should be taken when performing the mechanical parameter estimation.

Important Note and Warning:

To get the best estimation of the mechanical system parameters, jerky short high speed motions are required. The slower, or more gradual, the movements in the parameter estimation, the lower the data quality and lower the accuracy of the mechanical system parameters. Higher and jerkier moves result in more accurate data quality and better estimated mechanical system parameters. In particular, for estimating the inertia value, which is the most important parameter in closed loop control, higher acceleration/deceleration movements result in better inertia estimation. Therefore, when mechanical parameter estimation is performed, the user must be aware of the system and how it will be affected when high acceleration/deceleration moves are used. Finding a right setting that will result in the best parameter estimation without putting too much strain to the mechanical system will require trial and error and fine tuning.

Step 1 – Mechanical Parameter Estimation Acknowledgement

After the Auto Detect button is selected, following screen will show.

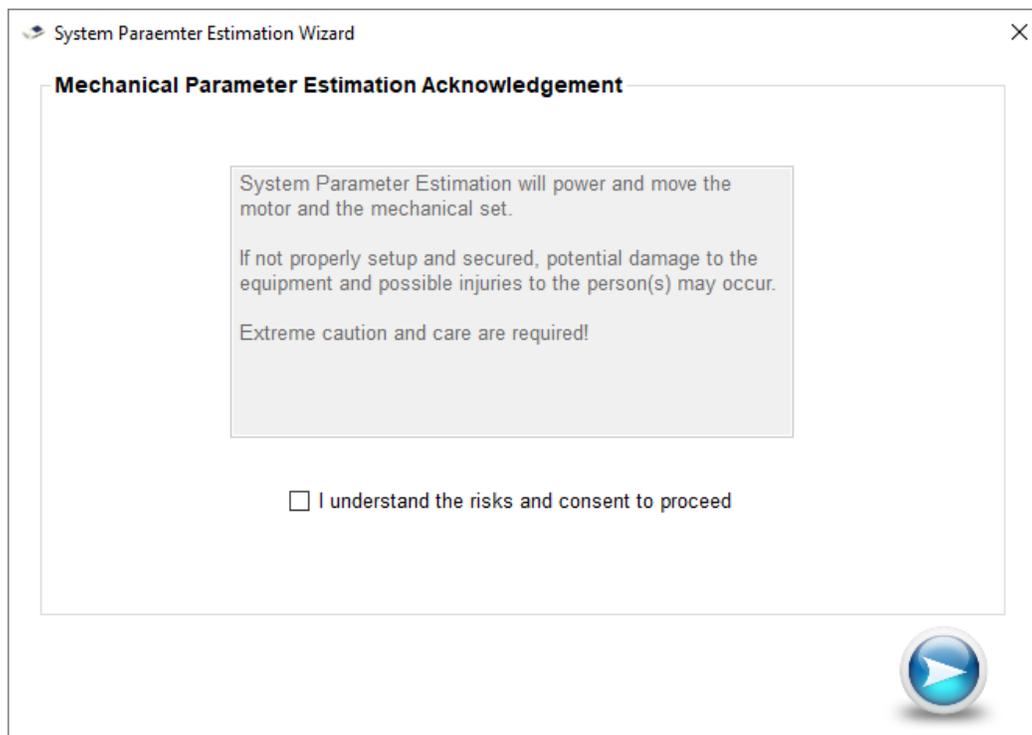


Figure 4.31

Check the acknowledgement checkbox to proceed to the next screen.

Step 2 – Mechanical Parameter Detection

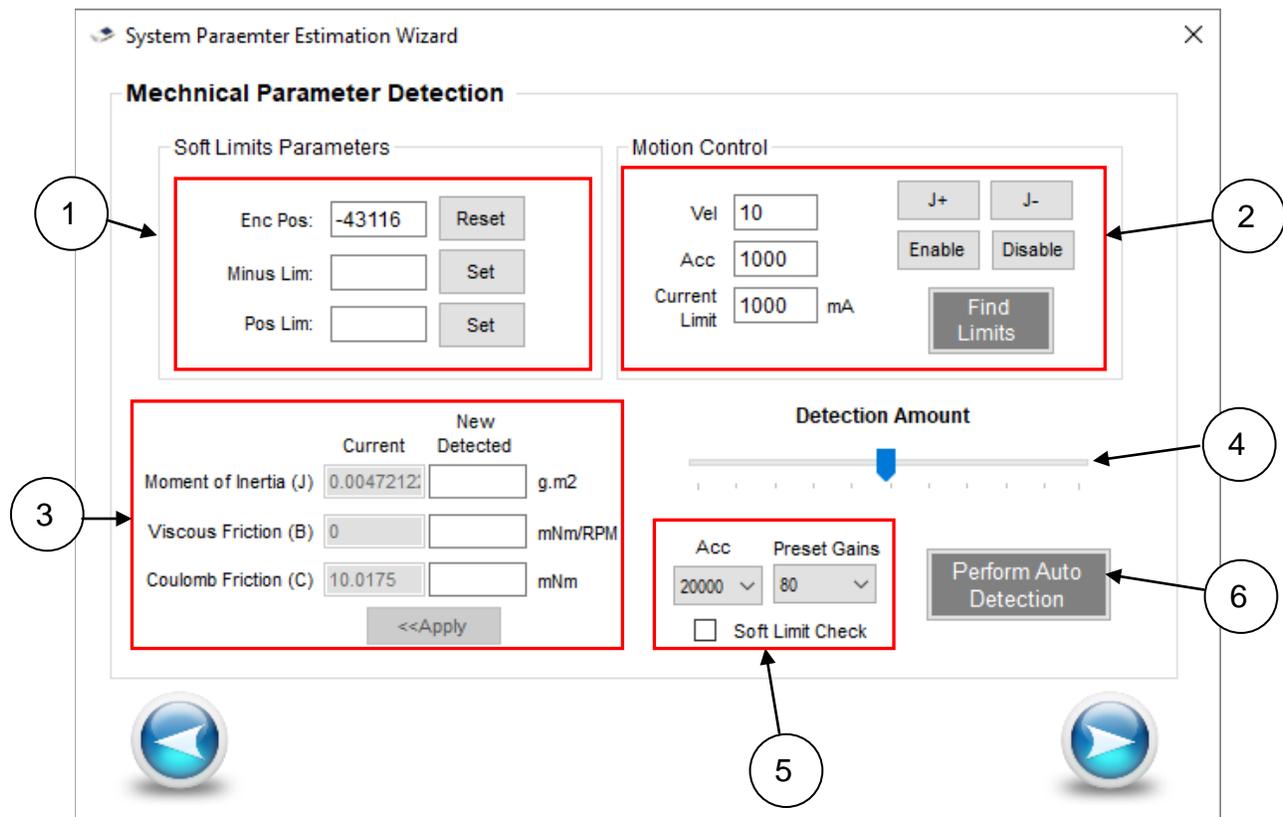


Figure 4.32

1. **Soft Limit Parameters:** If the mechanical system is a linear motion system, soft limits can be set to limit the travel during the mechanical estimation routine.
 - a. *Reset:* Sets the encoder counter to zero
 - b. *Minus Limit Set:* Defines the current position as minus limit
 - c. *Plus Limit Set:* Defines the current position as plus limit

There are two ways to set the soft limits. One method is to move the system manually by hand and setting the positions as negative and positive limits. Another method is to enable and jog the motor to the locations to set the soft limits.

Make sure to check the Soft Limit Check checkbox (See (5)) if soft limits are to be used.
2. **Motion Control:** With the default system parameters, system can be enabled and jogged to move the system to a safe location. Make sure to use a slow speed and current when moving. The system may not have proper mechanical system parameters and moves may not be stable. Preset gain may also need to be adjusted to jog the system.
3. **Current moment of inertia, viscous friction, and coulomb friction values** are shown. Newly detected parameters will show after the auto detection routine is successfully performed. Once new parameters are detected, user can use the new values by selecting the Apply button. Once the Apply button is

- selected, the new parameters will be used immediately for all movements, including the jog commands, and auto detection routines.
4. *Detection Amount:* This defines the movement amounts that will be used during the auto detect routines. The lower the value, the shorter the movements will be. When starting for the first time, use the lower values first and gradually increase until consistent parameters are detected. This will ease unforeseen stress on the system.
 5. Acceleration and Preset Gains can be set for the movements when doing the auto detection routines. If soft limit is to be used to limit the movements during the auto detect, check this box and make sure that the minus and positive limit values are set.
 6. *Perform Auto Detection:* Use this button to start the auto detection routine to get the estimated mechanical system parameters. The following dialog box will show when this Perform Auto Detect button is pressed.



Figure 4.33

Note: Anytime during the auto detect routine, the Abort button can be pressed to stop the motion and auto detect routine.

Once the auto detect mechanical system parameter estimation is done successfully, the new values will show on the *New Detected* fields.

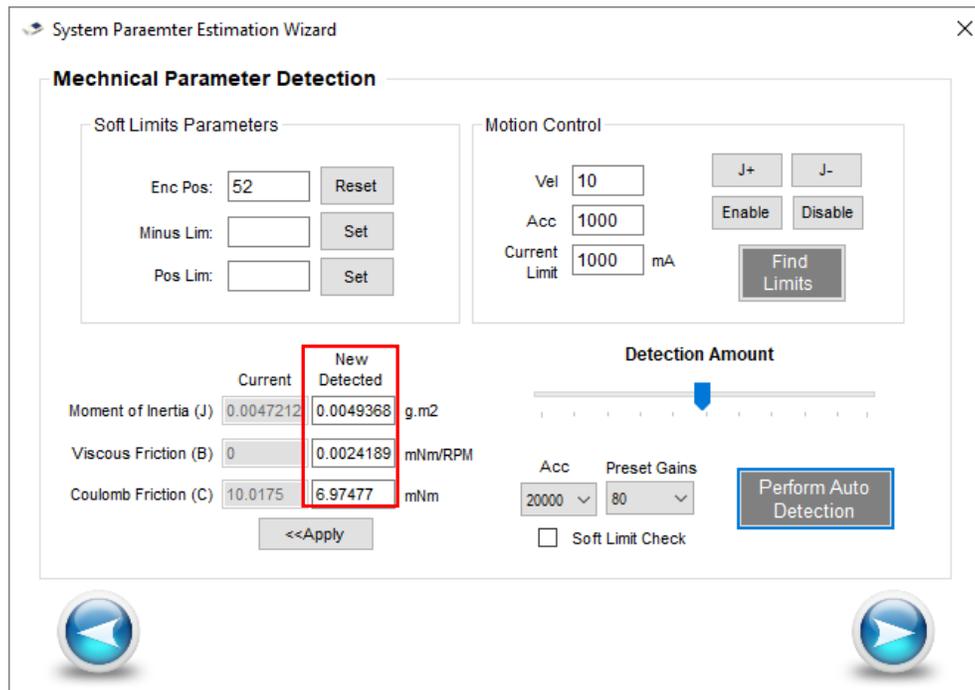


Figure 4.34

Review the newly detected parameters and press *Apply* button to use the new parameters.

It is recommended to perform the Auto Detect multiple times to get more accurate parameter estimations since better parameters will be used in each successive Auto Detect routines.

Once parameter estimation is done enough to get a satisfactory set of system mechanical parameters, click the Next page button. Following screen will show with the new parameters.

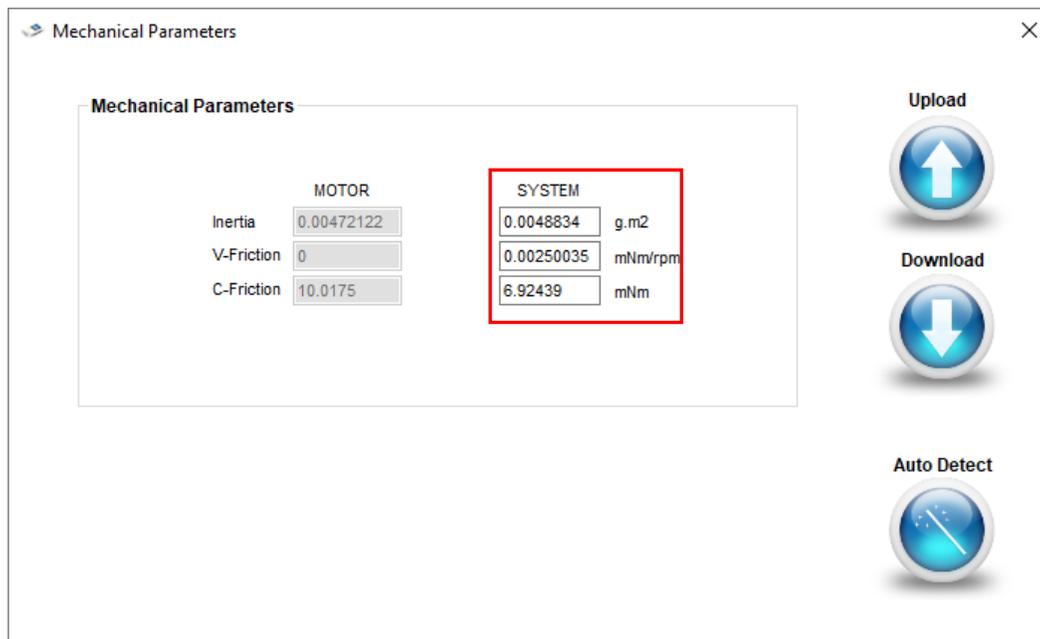


Figure 4.35

The values can be manually changed and downloaded if necessary, but not recommended unless you are aware of the effect that will have on the servo control performance.

For example, if you manually increase the inertia than the actual, control loop will assume a heavier mass and may try to over drive and compensate than necessary. If you manually decrease the inertia than the actual, control loop will assume a lighter mass and may result in lighter and slower response.

4.7. Configuration



Figure 4.36

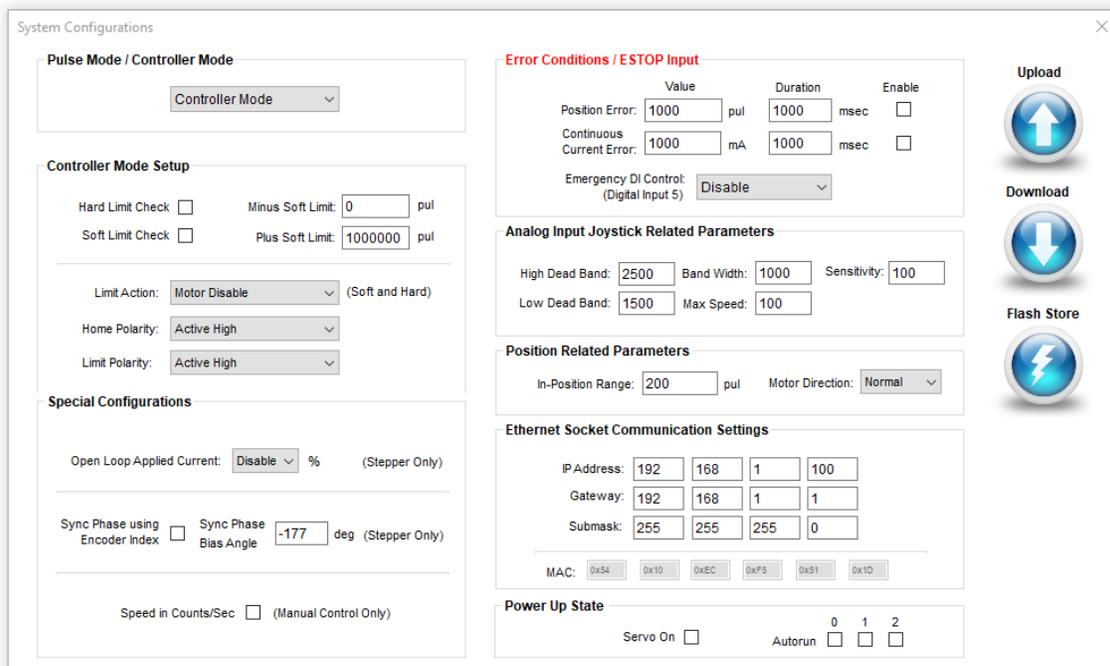


Figure 4.37

4.7.1. Mechanical Parameters

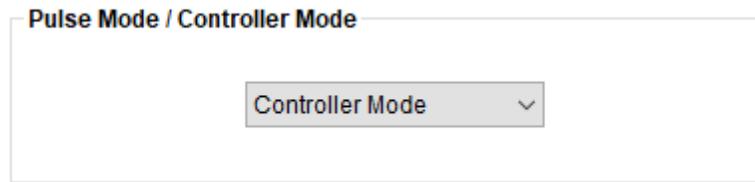


Figure 4.38

1. *Pulse Mode / Control Mode:* TITAN-SVX can be set as Controller Mode or Pulse Mode. In Controller Mode, the motion control and motion profiling is done via RS485 communication. In Pulse mode, the TITAN-SVX will accept pulse/direction or CW/CCW signals to determine motion.

4.7.2. Controller Mode Setup

If the TITAN-SVX is being set up in Controller Mode, various controller settings can be made.

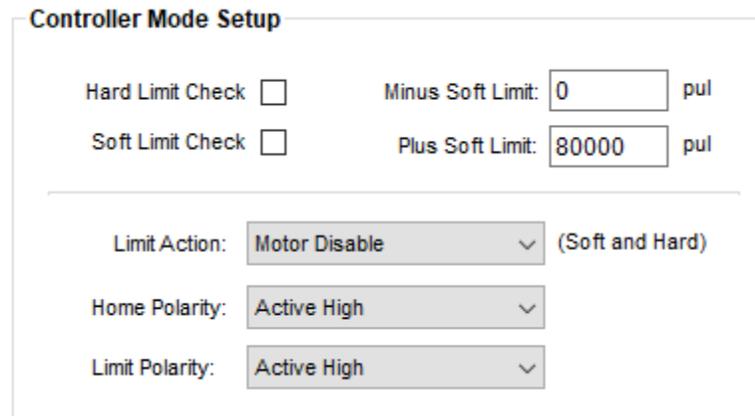


Figure 4.39

1. *Hard Limit Check:* When this is checked, the plus and minus limit inputs will be monitored. If the limit is triggered during the motion in the direction of the limit, the Limit Action will be performed.
2. *Soft Limit Check:* soft limit checks the position counter and performs limit checking. Plus and minus position limit values are set for the soft limit checking.
3. *Limit Action:* When hard or soft limit is triggered in the direction of the limit, limit action can be set to perform a desired action.
4. *Home Polarity:* Home digital input polarity can be set.
5. *Limit Polarity:* Limit digital input polarity can be set.

4.7.3. Pulse Mode Setup

If the TITAN-SVX is being set up in Pulse Mode, various controller settings can be made.

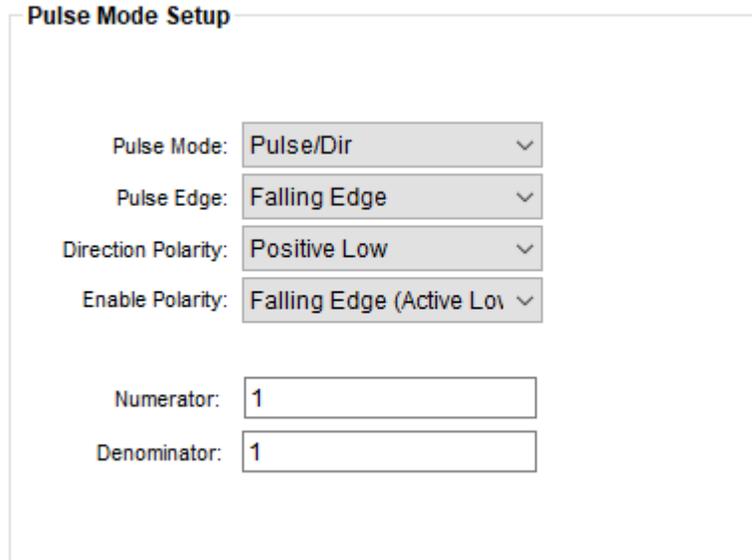


Figure 4.40

1. *Pulse Mode*: In Pulse Mode, one clock or two clock mode can be set. One Clock mode is Pulse/Dir mode and Two Clock Mode is CW/CCW mode.
2. *Pulse Edge*: Pulse rising or falling edge can be set.
3. *Direction Polarity*: Direction signal polarity can be set
4. *Enable Polarity*: Enable signal polarity can be set
5. *Numerator/Denominator*: Electronic Gearing is available to set the multiplication factor between the pulse and the encoder for movements. For example, if Numerator is set to 2 and Denominator is set to 1, the ratio value is 2 which will result in target moves of 2 encoder count moves per every pulse input.

4.7.4. Special Configurations

Special Configurations

Open Loop Applied Current: Disable % (Stepper Only)

Sync Phase using Encoder Index Sync Phase Bias Angle -177 deg (Stepper Only)

Speed in Counts/Sec (Manual Control Only)

Figure 4.41

1. *Open Loop Applied Current:* This setting applies only to the stepper motors. When this is enabled and set to a value other than Disable, TITAN-SVX goes to open loop control when motion is not being performed. When moving, closed loop servo control is performed. The defined percentage of the current means the percent of the maximum current value set in the motor parameter current setting. This feature is useful for highly sensitive optical applications when the motor must be still and not jittering when stationary. When full servo is performed at stationary, the motor may jitter and vibrate which may be undesirable. Note that when this feature is enabled, constant current value will be input to the coil of the stepper and will heat up the stepper motor, just as in open loop stepper control.
2. *Sync Phase:* When the motor enables, it will use the index position to correctly phase the motor. This is helpful when the motor enables with a heavy load attached. This feature is for stepper motors only.
3. *Speed in Counts/Sec:* Speed can be changed from units of RPM to encoder counts/sec. This will allow the motor to move at slower target speeds.

4.7.5. Error Conditions / ESTOP Input

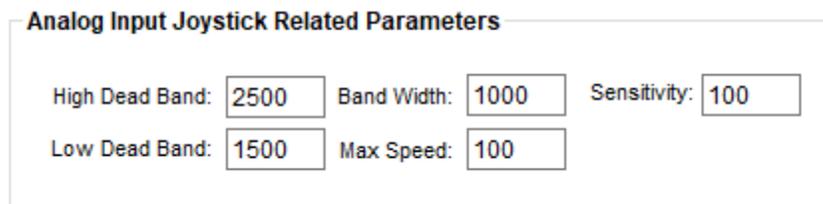
Error Conditions / ESTOP Input

	Value		Duration		Enable
Position Error:	1000	pul	1000	msec	<input type="checkbox"/>
Continuous Current Error:	1000	mA	1000	msec	<input type="checkbox"/>
Emergency DI Control: (Digital Input 5)	Disable				

Figure 4.42

1. Error Condition/ESTOP Input Settings
 - a. *Position Error and Duration*: If enabled and the position error is greater than the set amount for the set duration, a fault condition will occur. The motor will disable and all motion will stop.
 - b. *Current Error and Duration*: If enabled and the current error is greater than the set amount for the set duration time, a fault condition will occur. The motor will disable and all motion will stop.
 - c. Digital Input 5 can be configured as emergency digital input.

4.7.6. Analog Input Joystick Related Parameters



The screenshot shows a configuration window titled "Analog Input Joystick Related Parameters". It contains five input fields arranged in two rows. The first row includes "High Dead Band" (2500), "Band Width" (1000), and "Sensitivity" (100). The second row includes "Low Dead Band" (1500) and "Max Speed" (100).

Figure 4.43

1. *Dead Band*: Define the high and low dead band area for the joystick control. When joystick feature is enabled, motion would not occur while the analog input is within the high and low dead band.
2. *Band Width*: Define the width band area.
3. *Max Speed*: Define the max speed used for joystick operation.
4. *Sensitivity*: Changes the responsiveness of the motor as the analog input voltage changes.

4.7.7. Position Related Parameters



The screenshot shows a configuration window titled "Position Related Parameters". It contains two controls: "In-Position Range" set to 200 pul, and "Motor Direction" set to Normal (indicated by a dropdown arrow).

Figure 4.44

1. *In-Position Range*: This parameter sets the allowable error for the motor to be considered in position. If the motor is in position, the "inpos" status will be set and the "inpos" digital output can be activated for the pulse mode.
2. *Motor Direction*: The direction polarity of the motor can be set.

4.7.8. Ethernet Socket Communication Settings

Ethernet Socket Communication Settings

IP Address:	192	168	1	100
Gateway:	192	168	1	1
Submask:	255	255	255	0

MAC: 0x00 0x00 0x00 0x00 0x00 0x00

Figure 4.45

1. *IP Address*: Define the IP address of the TITAN-SVX
2. *Gateway*: Define the gateway address.
3. *Submask*: Define the submask.
4. *MAC Address*: The MAC address is displayed here. This is a read only value.

4.7.9. Power Up State

Power Up State

Servo On	<input type="checkbox"/>	Autorun	0	1	2
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 4.46

1. *Power Up State*: When TITAN-SVX is powered, following can be enabled:
 - a. Servo On
 - b. Standalone programs 0, 1, and 2

4.8. Parameters



Figure 4.47

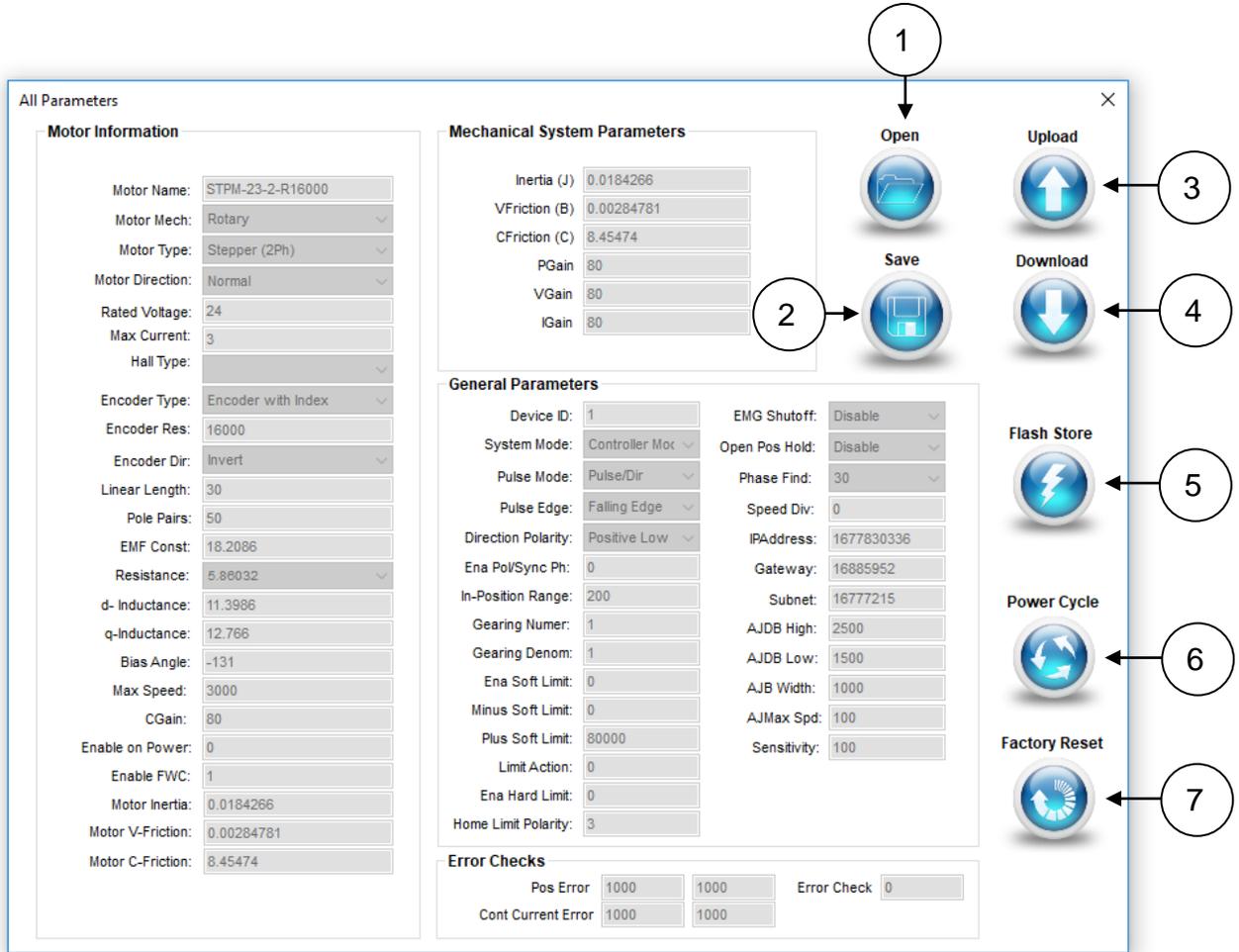


Figure 4.48

All setup and parameters on the TITAN-SVX are displayed in this screen. This includes all the motor parameters and mechanical system parameters. All configuration values are displayed as well. Note that all the values are read only.

System Parameter files are located in the following directory by default.

C:\Users\Public\Documents\TITAN-SVX-V3\SystemDB

1. Open System Parameter File
2. Save System Parameter File
3. Upload System Parameters from TITAN-SVX
4. Download System Parameters to TITAN-SVX
5. Flash Store – Stores the parameters to the flash memory of TITAN-SVX
6. Power Cycle – performs soft power cycle.
7. Factory Reset – resets the parameters to factory default.

4.9. Test Drive



Figure 4.49

The Test Drive screen will change depending on whether the TITAN-SVX is configured for Controller Mode or Pulse Mode. Note that in pulse mode, all motion control is disabled.

4.9.1. Control Mode

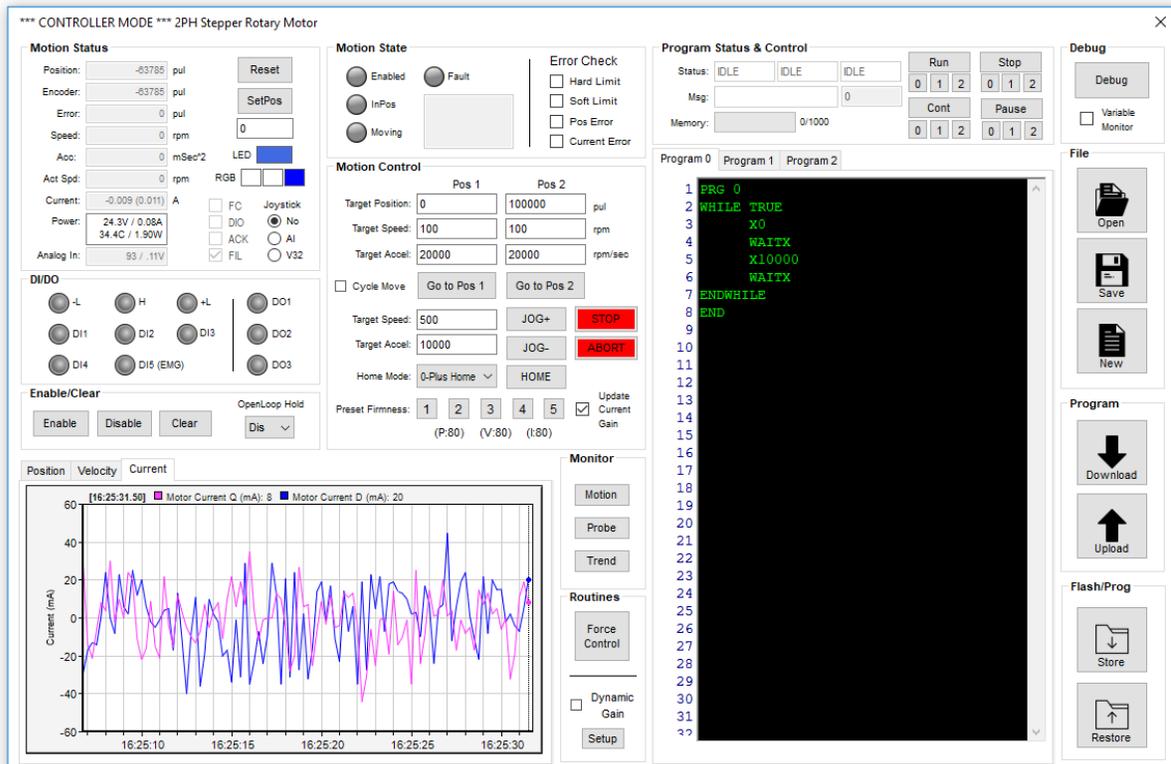


Figure 4.50

Motion Status

In the motion status section, following items are displayed

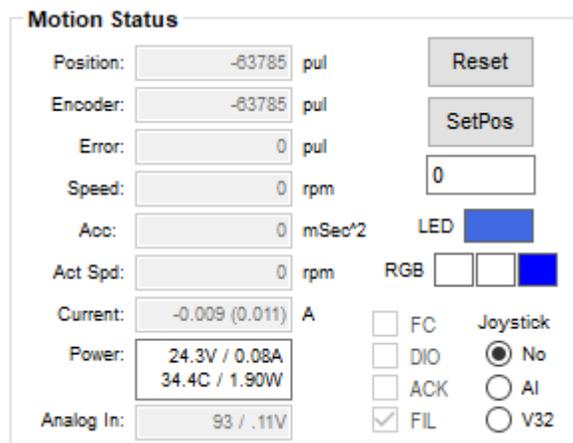


Figure 4.51

1. *Position:* Target position counter.
2. *Encoder:* Actual encoder position counter.
3. *Error:* The difference between the target Position and Encoder counter.
4. *Speed:* The target speed derived from of the motor

5. *Acceleration*: The current acceleration of the motor
6. *Actual Speed*: The actual speed derived from the encoder counter change
7. *Current*: The actual current of the motor winding
8. *Power*: Supply voltage and current
9. *Analog In*: The current status of the analog input
10. *LED*: The power LED status can be changed by clicking on LED indicator
11. *RGB*: The internal LED status can be changed by clicking on the RGB indicators
12. *FC*: Read only. Indicates if force control is enabled or disabled.
13. *DIO*: Read only. Indicates if force control uses DIO control.
14. *ACK*: Read only. Indicates if force control uses an acknowledgement signal.
15. *FIL*: Read only. Indicates if force control current filter is enabled or disabled.
16. *Joystick*: Enable or disable the joystick control feature. The joystick operation can use the analog input or the value within standalone variable V32.

Digital IO

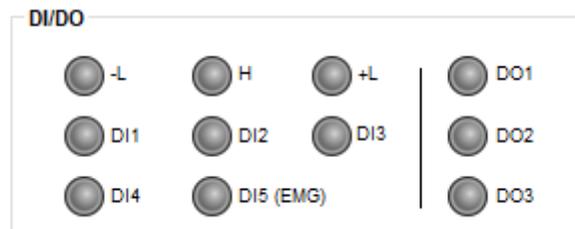


Figure 4.52

1. *Digital Input Status*: Displays status of all available digital inputs including – L/H/+L inputs and DI1 through DI5
2. *Digital Output Status*: Displays status of DO1 through DO3. Digital output can be toggled by clicking on the DO indicators.

Enable/Clear

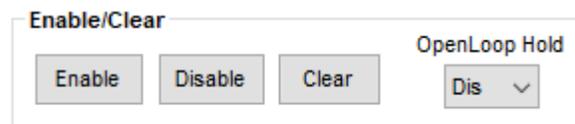


Figure 4.53

1. *Enable*: Motor servo on
2. *Disable*: Motor servo off
3. *Clear*: Fault condition is cleared
4. *Open Loop Hold*: The open loop position hold is disabled or enabled at the chosen percentage value.

Motion State



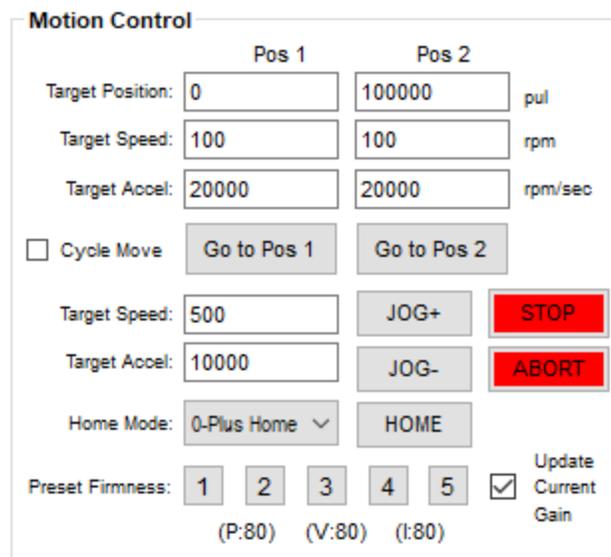
The Motion State control panel includes the following elements:

- Motion State:** Three radio buttons for 'Enabled', 'InPos', and 'Moving', and one radio button for 'Fault'.
- Error Check:** Four checkboxes for 'Hard Limit', 'Soft Limit', 'Pos Error', and 'Current Error'.

Figure 4.54

1. *Enabled:* Displays if servo is on or off
2. *InPos:* Displays if encoder position is within the in position setting of the target position.
3. *Moving:* Indicates whether the motor is moving or not from the encoder position change
4. *Fault:* The TITAN-SVX is in fault state. The fault description is listed if applicable.
5. *Error Check:* Enable or disable the following error conditions
 - a. Hard Limit
 - b. Soft Limit
 - c. Position Error
 - d. Current Error

Motion Control



The Motion Control panel includes the following elements:

- Target Position:** Input fields for Pos 1 (0) and Pos 2 (100000) with units (pul).
- Target Speed:** Input fields for Pos 1 (100) and Pos 2 (100) with units (rpm).
- Target Accel:** Input fields for Pos 1 (20000) and Pos 2 (20000) with units (rpm/sec).
- Cycle Move:** A checkbox and two buttons: 'Go to Pos 1' and 'Go to Pos 2'.
- Target Speed:** Input field (500) and buttons: 'JOG+', 'STOP'.
- Target Accel:** Input field (10000) and buttons: 'JOG-', 'ABORT'.
- Home Mode:** A dropdown menu (0-Plus Home) and a 'HOME' button.
- Preset Firmness:** Buttons for 1, 2, 3, 4, 5 with units (P:80), (V:80), (I:80) and an 'Update Current Gain' checkbox.

Figure 4.55

1. *Pos 1 and Pos2:* Dual Target Move Position/Velocity/Acceleration Settings. The Go to Pos 1/2 buttons can be used to move the motor to the defined positions using the defined speed settings.
2. *Cycle Mode:* Continuously move between the dual target positions
3. *JOG+/JOG-:* jog in the minus or plus direction
4. *HOME:* Perform the home routine chosen from the Home Mode dropdown menu.

5. *Preset Gains*: Sets the gains to one of the 5 preset values. Following gain values are set when one of the preset gain buttons is press
 - a. Position Gain
 - b. Velocity Gain
 - c. Integral Gain
6. *Current Gain* (if the current gain box is checked)
7. *Update Current Gain*: Include the current gain when using the preset firmness options. The current gain can also be updated independently through the Motion option.

Monitor



Figure 4.56

1. **Motion Monitor:** When this button is pressed, the following dialog box will open which shows the target/actual position on the first graph, target/actual velocity on the second graph, and target/actual current on the third graph.

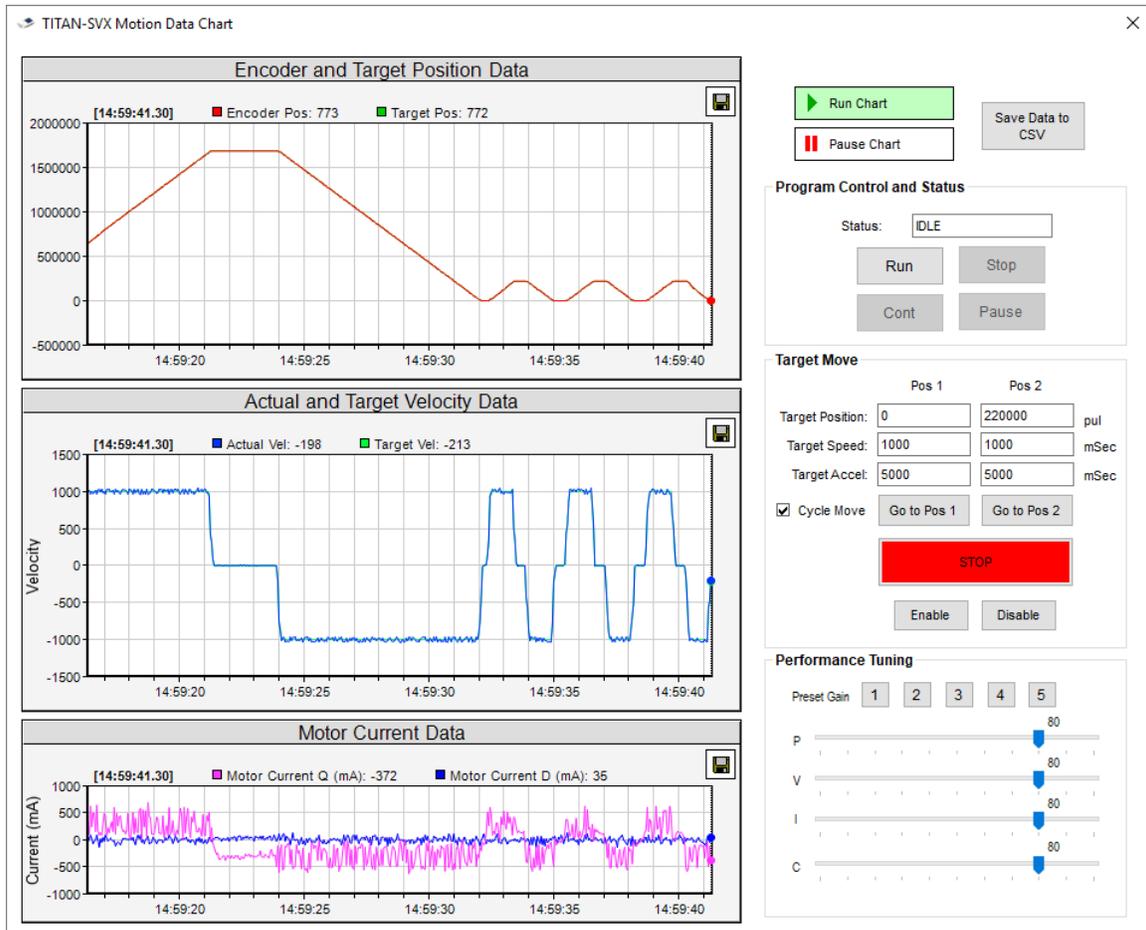


Figure 4.57

In this Motion Monitor screen, following can be done:

- a. Motion graphs can be saved to CSV file
- b. Motion program can be run
- c. Dual motion movement test can be done

- d. Motor can be enabled and disabled
 - e. Individual gains can be set
 - f. One of the five preset gains can be set.
2. *AccuProbe*: Gather an accurate collection of the motion data. The following screen will show when Probe button is pressed. Three data sets can be selected for probing. Each data set consists of 250 points.

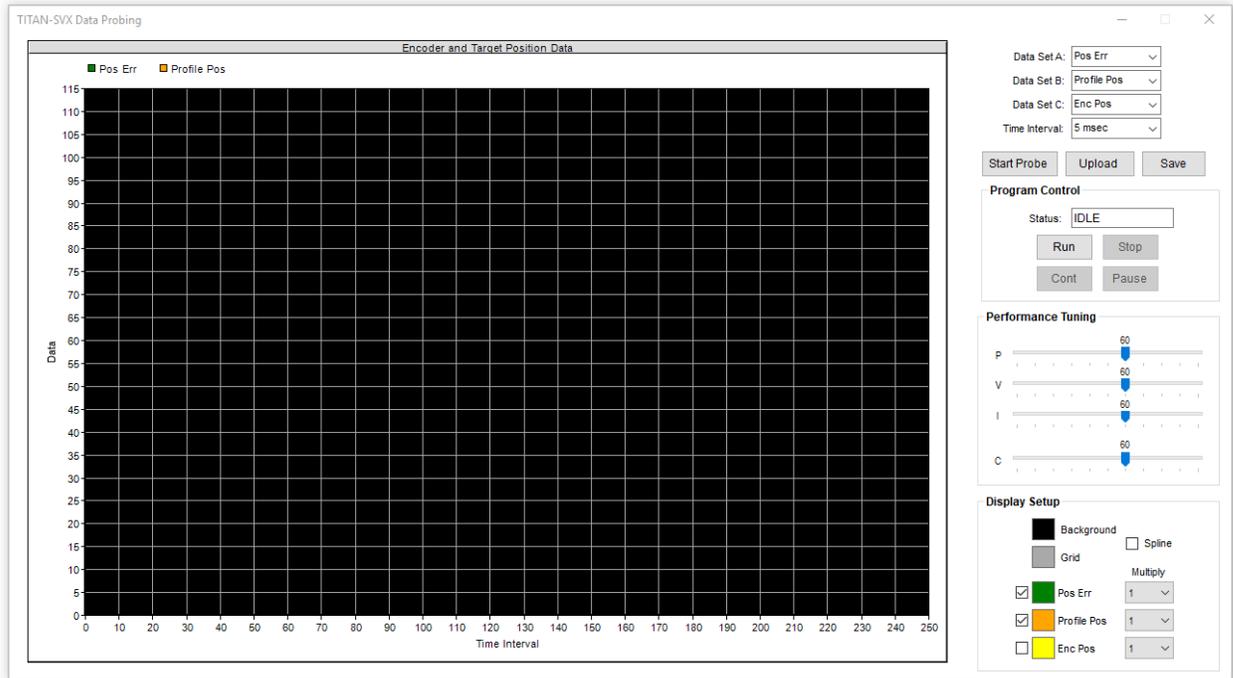


Figure 4.58

From the Probe screen following can be done

- a. Three sets of data can be collected. A data set can be any of the following: Encoder Position, Encoder Velocity, Profile Position, Profile Velocity, Position Error, Velocity Error, Actual Current, and Target Current.
- b. Following probe duration can be set: 1, 2, 5, 10, 20, 50 msec.
- c. Start Probe can be done
- d. Probe data sets can be stored to CSV file.
- e. Program control can be done
- f. Gain settings can be done.
- g. Display settings can be configured.

3. *Trend*: Trend values are displayed. Following screen will show when Trend button is pressed.

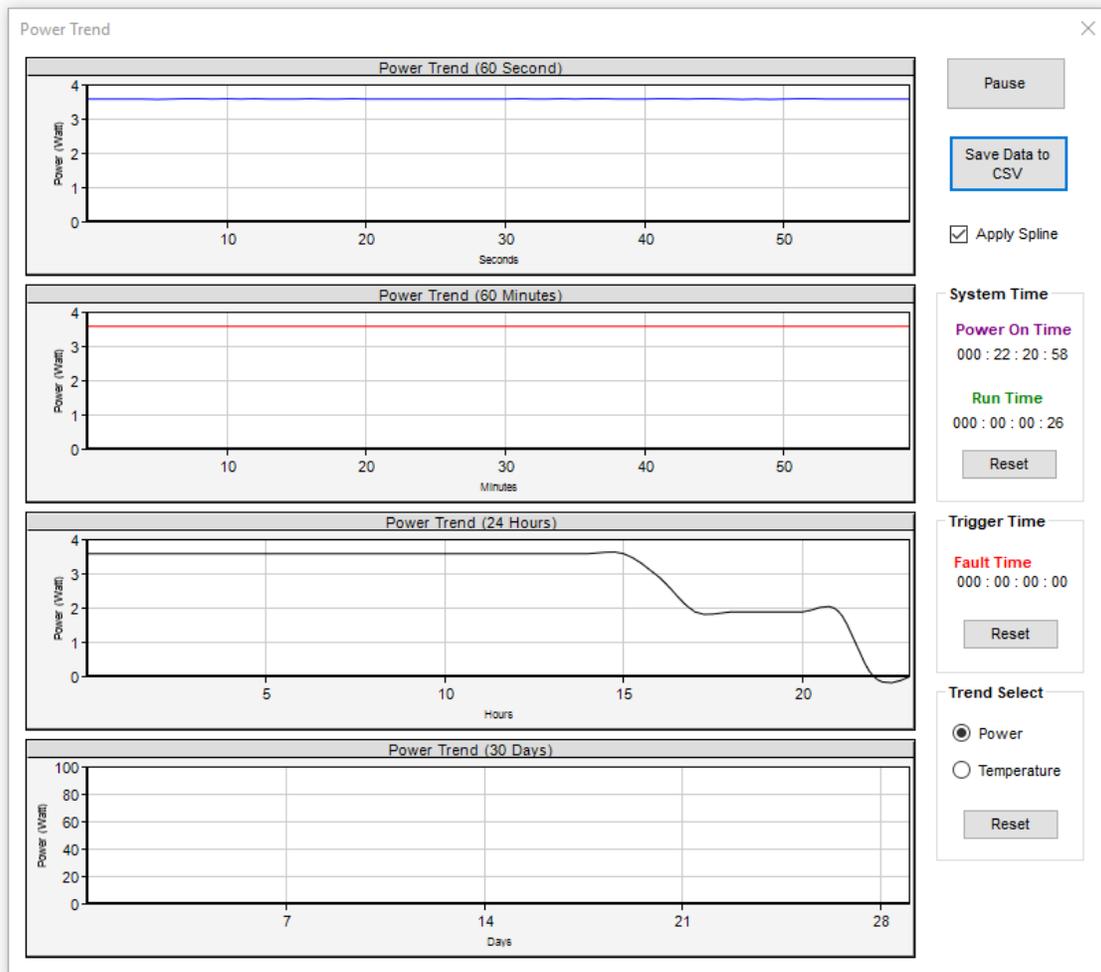


Figure 4.59

- a. First graph shows the previous 60 second trend values
- b. Second graph shows the previous 60 minute trend values
- c. Third graph shows the previous 24 hour trend values
- d. Fourth graph shows the previous 30 day trend values
- e. Trend values are reset at power cycle.
- f. Trend values can be saved to a CSV file
- g. System Time shows the total time TITAN-SVX power was one and total time motor was enabled and moving.
- h. Trigger time for the Fault is shown with respect to the System Time
- i. Trend of Power or Temperature can be set. Note that once the trend type is changed, the values are reset.

Routines



Figure 4.60

1. **Force Control:** When the Force Control button is pressed the following screen will appear to provide access to the Force Control feature.

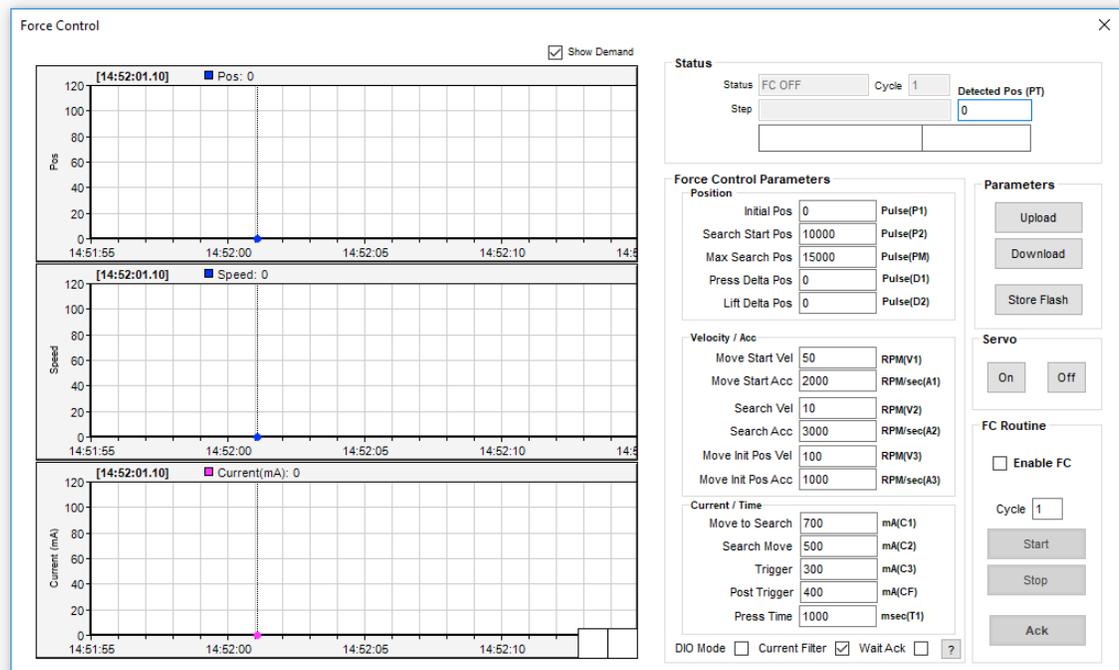


Figure 4.61

All parameters for the Force Control feature can be entered here. In order for any new parameters to take effect, click the Download button.

The Upload button will pull to current Force Control values off the TITAN-SVX and populate the corresponding fields.

A Force Control routine can be saved to the flash memory of a TITAN-SVX with the Store Flash button.

2. *Dynamic Gain*: The check box for the Dynamic Gain can be used to enable or disable the Dynamic Gain feature. When the Setup button is pressed following screen will show.

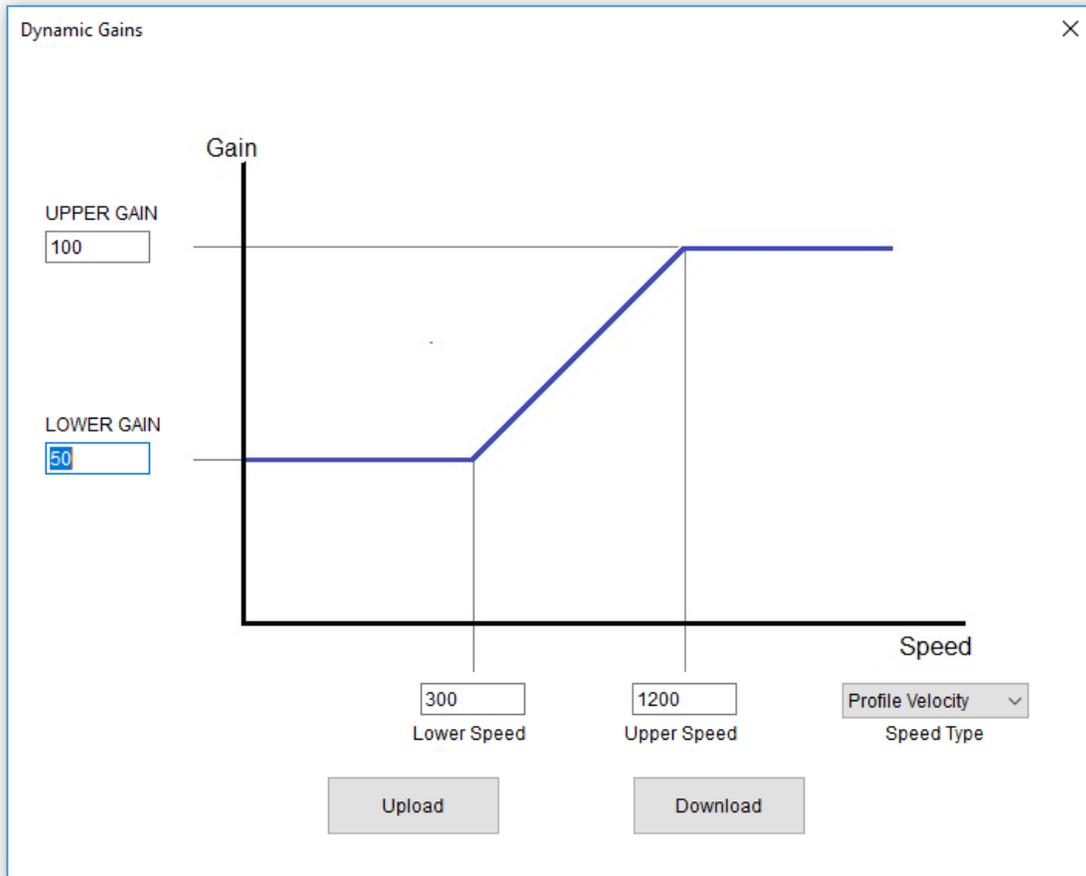


Figure 4.62

Upper gain, lower gain, upper speed, and lower speed can be defined in this window. Speed value that will be used for dynamic gain can be either the profile velocity or the actual encoder velocity. Upload and download the dynamic gain parameters.

Standalone Programming Environment



Figure 4.63

1. **Program Status & Control:** TITAN-SVX supports multi-threaded programming of up to 3 threads. The status of each thread is shown. The total size of programs (all three threads) is 1000 lines.
2. **Run/Stop/Cont/Pause:** Program can be controlled individually or at the same time. Run will run the program from the beginning. Stop will stop the program that is running or paused. Pause will pause the program that is running. Continue will continue the program that is paused.
3. **Debug:** TITAN-SVX has program debugging capability. Note that only one program can be debugged at a time. When the Debug button is pressed, the program that is open will be compiled and downloaded for debugging. Debugging screen will show when debug button is pressed.

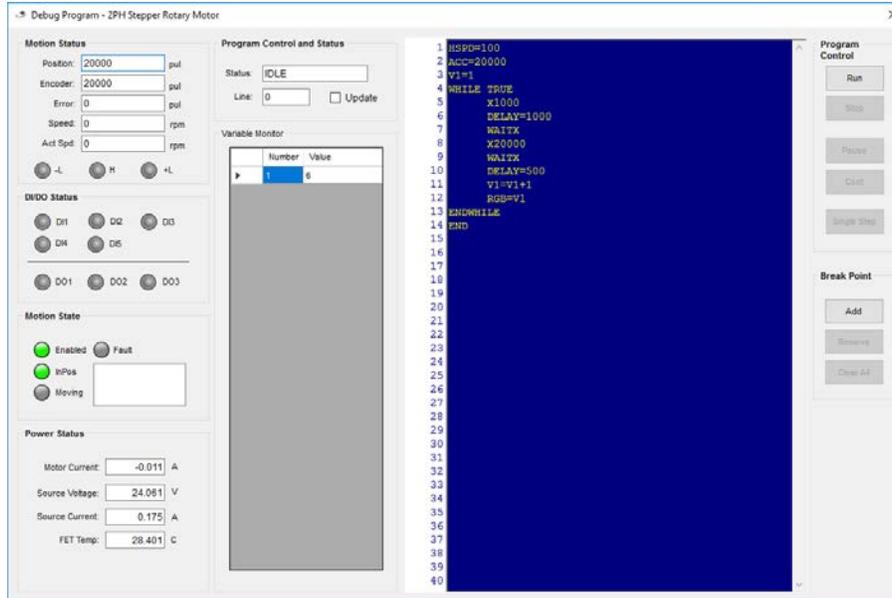


Figure 4.64

Break points, single stepping, and variable monitoring are supported in debugging.

4. *Variable Monitor*: Variable Monitor is selected, following screen will show.



Figure 4.65

Variables can be added, deleted, and edited from the variable monitor list.

5. *File*: Program files can be saved, opened, and cleared for new program. TITAN-SVX standalone programs are stored in the following directory:

C:\Users\Public\Documents\TITAN-SVX-V3\MotionPrograms

6. **Download/Upload:** Programs are compiled and downloaded to the TITAN-SVX. When Compile Download button is pressed, all three threads are compiled and downloaded. The program in the TITAN-SVX can also be decompiled and uploaded to the program editor.
7. **Store/Restore:** Programs can be compiled and stored in the flash memory for permanent storing of the program on the TITAN-SVX. Program can also be uploaded from the flash memory and decompiled to the program editor.
8. **Program Editor:** Program editor is available to write, modify the programs. Note the tabs on the top of the program edit to select different program thread.

Details of the Standalone programming language will be explained in a separate section of the manual.

4.9.2. Pulse Mode

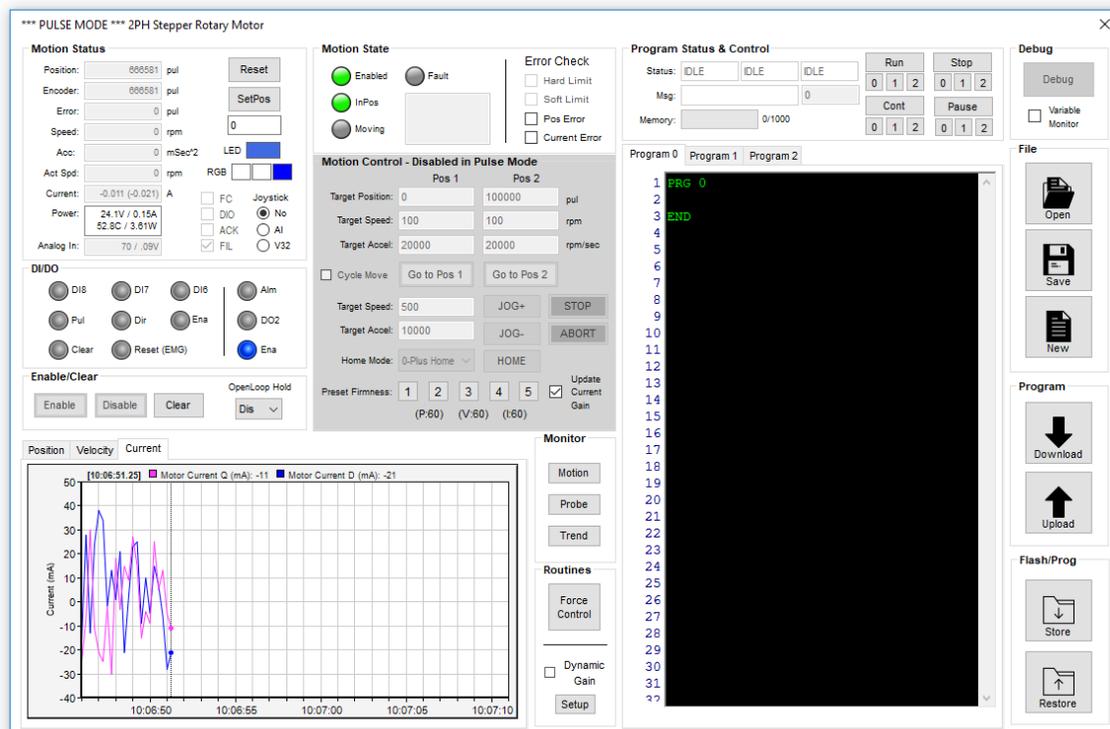


Figure 4.66

In Pulse mode, the Motion Control portion of the Test Drive section is disabled. Motion can only be achieved via the pulse/direction or CW/CCW signals.

Contact Information

Arcus Servo, Inc.

3159 Independence Drive
Livermore, CA 94551
925-373-8800

www.arcusservo.com

The information in this document is believed to be accurate at the time of publication but is subject to change without notice

